

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφορικά και
Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή



Ανάπτυξη μιας Εφαρμογής Δικτυακού Παιχνιδιού με
Εκπαιδευτικά Χαρακτηριστικά σε Πλατφόρμα Κινητού
Υπολογισμού και Αξιολόγησή της.

Πέτρος Καστρινάκης

Επιβλέπων Καθηγητής
Χρήστος Γκουμόπουλος

Σεπτέμβριος 2015

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

**Μεταπτυχιακό Πρόγραμμα Σπουδών Πληροφοριακά και
Επικοινωνιακά Συστήματα**

Μεταπτυχιακή Διατριβή

**Ανάπτυξη μιας Εφαρμογής Δικτυακού Παιχνιδιού με
Εκπαιδευτικά Χαρακτηριστικά σε Πλατφόρμα Κινητού
Υπολογισμού και Αξιολόγησή της.**

Πέτρος Καστρινάκης

**Επιβλέπων Καθηγητής
Χρήστος Γκουμόπουλος**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Πληροφοριακά και Επικοινωνιακά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Σεπτέμβριος 2015

Περίληψη

Τα παιδιά σήμερα από την προσχολική ηλικία και κατά τη διάρκεια όλης της σχολικής τους περιόδου, έχουν πρόσβαση στη χρήση ψηφιακών εργαλείων και τεχνολογίας ως σημαντικό μέρος της καθημερινής τους ζωής. Κατά συνέπεια αποκτούν από νωρίς εξοικείωση με τα ψηφιακά εργαλεία, στα οποία συμπεριλαμβάνονται και οι κινητές πλατφόρμες (έξυπνα κινητά τηλέφωνα, tablets, κ.τ.λ.). Ενώ έχει ήδη αναπτυχθεί μια πληθώρα εφαρμογών γι' αυτές τις πλατφόρμες, υπάρχει σημαντική έλλειψη λογισμικών με εκπαιδευτικό χαρακτήρα και κυρίως με υποστήριξη πολλών παικτών ταυτόχρονα (multiplayer) από διαφορετικές κινητές πλατφόρμες σε τοπικό επίπεδο. Επιπλέον, η πλειοψηφία των διαθέσιμων εφαρμογών δεν είναι στην ελληνική γλώσσα.

Στόχος αυτής της διατριβής είναι ο σχεδιασμός, η υλοποίηση και η αξιολόγηση μιας δικτυακής εφαρμογής-παιχνιδιού σε πλατφόρμα κινητού υπολογισμού με εκπαιδευτικά χαρακτηριστικά, χρησιμοποιώντας τις αρχές της αντικειμενοστρεφούς τεχνολογίας. Η εφαρμογή έχει ψυχαγωγικό, συνεργατικό και εκπαιδευτικό χαρακτήρα, προσπαθώντας παράλληλα να αξιοποιήσει τις δυνατότητες που παρέχουν οι σύγχρονες πλατφόρμες κινητού υπολογισμού (όπως για παράδειγμα η αίσθηση του περιβάλλοντος μέσω αισθητήρων, κτλ), με υποστήριξη της ελληνικής γλώσσας. Η εφαρμογή θα χρησιμοποιεί, είτε σύνδεση Bluetooth, είτε Wi-Fi για τη σύνδεση μεταξύ των συσκευών και η διεπαφή θα παρέχει υποστήριξη της ελληνικής γλώσσας.

Ο «Λαβύρινθος», το παιχνίδι για φορητές συσκευές Android που επιχειρείται να αναπτυχθεί, αφορά λαβύρινθους με τυχαία μονοπάτια. Υπάρχουν πολλές διαφορετικές διαδρομές, όπου εμφανίζονται μερικές Σφίγγες με αινίγματα πληροφορικής επιπέδου πέμπτης – έκτης δημοτικού και διαφορετικό χρώμα, ανάλογα με την ερώτηση που θα κάνουν. Ο παίκτης επιλέγει και πέφτει κάθε τόσο πάνω σε μια Σφίγγα, όπου για να περάσει θα χρειαστεί να απαντήσει σωστά στην ερώτησή της. Οι ερωτήσεις μπορεί να είναι πολλαπλών-επιλογών ή τύπου πίνακα και διαφορετικών επιπέδων δυσκολίας και κατηγορίας. Σκοπός του παιχνιδιού είναι να καταφέρει ο παίκτης να βρει την έξοδο από το λαβύρινθο. Στο ομαδικό μέρος του παιχνιδιού, οι παίκτες μπορούν να συναντηθούν και να συνεχίσουν από κοινού την προσπάθεια στα μονοπάτια να απαντήσουν σωστά στα αινίγματα των Σφιγγών και να βρουν τελικά την έξοδο.

Το ερευνητικό ερώτημα που τίθεται είναι κατά πόσο η χρήση ενός τέτοιου ψυχαγωγικού παιχνιδιού με εκπαιδευτικό προσανατολισμό μπορεί να βοηθήσει την διαδικασία μάθησης και την ανάπτυξη δεξιοτήτων.

Summary

Children today from pre-school through their entire school age, have access to the use of digital tools and technology as an important part of their daily lives. Consequently they gain early familiarity with the digital tools, including and mobile platforms (smartphones, tablets, etc.). While there have already been developed numerous applications for these platforms, there is a significant lack of educational software, especially with the support of many players simultaneously (multiplayer) from different mobile platforms locally. Moreover, the majority of available applications is not in Greek language.

The aim of this thesis is the design, implementation and evaluation of a multiplayer game application in a mobile computing platform with educational features, using the principles of object-oriented technology. The application has entertainment, collaborative and educational characteristics, while trying to exploit the possibilities offered by modern mobile computing platforms (for example the sense of the environment through sensors, etc.), with support of the Greek language. The application will use either Bluetooth, or Wi-Fi for the connection between the devices and the interface will support the Greek language.

“Labyrinth”, the game for Android mobile devices which is going to be developed, has to do with moving across labyrinths with random paths. There are many alternative routes, where Sphinxes with riddles of level fifth to sixth grade appear, and each one with a different color, depending on the question they will pose. The player chooses a route and falls every now and then on a sphinx, to which should answer the question in order to pass. Questions may be multiple-choice, or table type with different levels of difficulty and category. The purpose of the game is to accomplish player to find his way out of the maze. In the cooperative part of the game, players can meet and continue together the effort through the paths to answer correct to the sphinxes’ riddles and finally find the way out.

The research question posed is whether the use of such an entertaining game with educational orientation can help the learning process and skills development.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Χρήστο Γκουμόπουλο, για την εμπιστοσύνη που μου έδειξε, την υποστήριξή του, αλλά και γιατί στάθηκε η αφορμή να ασχοληθώ με τον σχεδιασμό και υλοποίηση εφαρμογής για κινητές συσκευές με λειτουργικό σύστημα Android.

Πολλές ευχαριστίες επιθυμώ να δώσω επίσης στους συναδέλφους μου εκπαιδευτικούς, για τη συμμετοχή και τη συνεργασία τους στην εκπόνηση της έρευνας.

Πάνω απ' όλα όμως, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου και ιδιαίτερα στη σύζυγό μου Ειρήνη, για τη βοήθεια και συμπαράσταση στις δύσκολες στιγμές, καθώς και για την υπομονή που επέδειξαν σε όλη την τριετή περίοδο των σπουδών μου στο Ανοικτό Πανεπιστήμιο Κύπρου.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ	1
1.1 ΣΚΟΠΟΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΕΥΝΑΣ	2
1.1.1 Αναγκαιότητα και σπουδαιότητα έρευνας	3
1.1.2 Τα μοντέλα των Gagne και Laurillard.	4
1.1.3 Προσδοκώμενα αποτελέσματα.....	6
1.1.4 Βασικά ερευνητικά ερωτήματα:	7
1.1.5 Συμπεράσματα.....	8
1.2 ΕΡΕΥΝΗΤΙΚΗ ΠΡΩΤΟΒΟΥΛΙΑ	8
1.2.1 Λειτουργικά συστήματα.....	8
1.2.2 Δικτυακά παιχνίδια.....	9
1.2.3 Χαρακτηριστικά Εφαρμογών/Παιχνιδιών σε κινητές πλατφόρμες με Android.	15

ΚΕΦΑΛΑΙΟ 2

ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ	16
2.1 ΜΕΘΟΔΟΛΟΓΙΑ	16
2.2 ΕΚΠΑΙΔΕΥΤΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	17
2.3 ΑΠΑΙΤΗΣΕΙΣ (ΣΥΓΚΕΝΤΡΩΣΗ ΤΩΝ ΖΗΤΟΥΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ)	18
ΛΑΒΥΡΙΝΘΟΣ	18
2.4 ΑΝΑΛΥΣΗ	21
2.4.1 Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram).....	22
2.4.2 Διάγραμμα Καταστάσεων (Statechart Diagram)	23
2.4.3 Διάγραμμα Δραστηριοτήτων (Activity Diagram).....	23
2.4.4 Διαγράμματα Κλάσεων (Class Diagrams).....	25
2.4.5 Διάγραμμα Συστατικών (Component Diagram)	42
2.4.6 Η Βάση Δεδομένων της Εφαρμογής.....	43

ΚΕΦΑΛΑΙΟ 3

Η ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ «ΛΑΒΥΡΙΝΘΟΣ»	46
3.1 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑΣ	47
3.1.1 Android SDK APIs	47
3.1.2 Εργαλεία και Περιβάλλον Ανάπτυξης	48
3.1.3 Απαιτήσεις συστήματος.....	49
3.2 Η ΔΙΕΠΑΦΗ ΣΥΣΤΗΜΑΤΟΣ	50
3.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΑΞΙΟΠΟΙΟΥΜΕΝΕΣ ΔΥΝΑΤΟΤΗΤΕΣ	55

ΚΕΦΑΛΑΙΟ 4

ΑΠΟΤΙΜΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ	57
4.1 ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΠΡΙΝ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	58
4.2 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΜΕΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	63
4.3 ΠΑΡΑΤΗΡΗΣΕΙΣ ΚΑΙ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΒΕΛΤΙΩΣΕΙΣ ΑΠΟ ΕΚΠΑΙΔΕΥΤΙΚΟΥΣ.....	72
4.4 ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	73

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΑ ΣΧΕΔΙΑ	75
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	75
5.1.1 Συμπεράσματα της Έρευνας.....	76
5.1.2 Σύνοψη.....	76
5.2 ΣΚΕΨΕΙΣ ΚΑΙ ΜΕΛΛΟΝΤΙΚΑ ΣΧΕΔΙΑ	78
5.3 ΕΠΙΛΟΓΟΣ.....	79

ΠΑΡΑΡΤΗΜΑ Α

ΕΡΩΤΗΜΑΤΟΛΟΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΩΝ	80
Α.1 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΠΡΙΝ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	80
Α.2 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΜΕΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	81

ΠΑΡΑΡΤΗΜΑ Β

ΑΡΧΕΙΑ XML	85
Β.1 ΤΟ ΑΡΧΕΙΟ MANIFEST	85

B.2 ACTIVITIES.....	86
B.3 RESOURCE REFERENCES	88
B.4 ΠΡΟΣΑΡΜΟΣΜΕΝΑ ΚΟΥΜΠΙΑ.....	89
ΠΑΡΑΡΤΗΜΑ Γ	
ΚΩΔΙΚΑΣ JAVA	90
Γ.1 APXEIO MAIN.JAVA.....	90
Γ.2 APXEIO PLATFORM.JAVA	92
Γ.3 APXEIO GAME.JAVA	95
Γ.4 APXEIO ASSETS.JAVA.....	100
Γ.5 APXEIO SPATIALHASHGRID.JAVA.....	102
Γ.6 APXEIO SYNDESI.JAVA	104
Γ.7 APXEIO BLUETOOTH.JAVA	108
Γ.8 APXEIO GRAFIKA.JAVA	114
Γ.9 APXEIO EMFANISI.JAVA	118
Γ.10 APXEIO UIBUTTON.JAVA	119
Γ.11 APXEIO TOUCHSCREEN.JAVA.....	120
Γ.12 APXEIO ACCELEROMETER.JAVA.....	121
Γ.13 APXEIO BASI.JAVA	122
Γ.14 APXEIO ENIGMA.JAVA.....	126
Γ.15 APXEIO EROTISI.JAVA	127
Γ.16 APXEIO APANTISI.JAVA.....	128
Γ.17 APXEIO STOIXEIO.JAVA.....	128
Γ.18 APXEIO PLAYER.JAVA	129
Γ.19 APXEIO SFIGGA.JAVA	132
Γ.20 APXEIO DIADROMI.JAVA	134
Γ.21 APXEIO EISODOS.JAVA	135
Γ.22 APXEIO EXODOS.JAVA	137
Γ.23 APXEIO BOX.JAVA.....	137
Γ.24 APXEIO SCORE.JAVA	138
Γ.25 APXEIO STATE.JAVA.....	139
Γ.26 APXEIO AVATAR.JAVA.....	140
Γ.27 APXEIO LABYRINTH.JAVA	141
Γ.28 APXEIO EIDOS.JAVA	151
Γ.29 APXEIO MULTIPLECHOICE.JAVA	153
Γ.30 APXEIO PINAX.JAVA.....	156
Γ.31 APXEIO BONUSFASI.JAVA	160
Γ.32 APXEIO ZARI.JAVA	166
Γ.33 APXEIO GAMEOVER.JAVA.....	168

Κεφάλαιο 1

Εισαγωγή

Σήμερα, τα παιδιά από την προσχολική ηλικία και κατά τη διάρκεια όλης της σχολικής τους περιόδου, έχουν πρόσβαση στη χρήση ψηφιακών εργαλείων και τεχνολογίας ως σημαντικό μέρος της καθημερινής τους ζωής. Κατά συνέπεια αποκτούν από νωρίς εξοικείωση με τα ψηφιακά εργαλεία, στα οποία συμπεριλαμβάνονται και οι κινητές πλατφόρμες (έξυπνα κινητά τηλέφωνα, tablets, κ.τ.λ.). Τα έξυπνα κινητά τηλέφωνα και οι ταμπλέτες χρησιμοποιούνται από τεράστιο πλήθος νέων ανθρώπων, ως μέρος της κοινωνικής και πολιτιστικής τους ζωής. Είναι πλέον σχετικά φθηνά και εύκολα στη χρήση.

Λειτουργικά συστήματα όπως τα Windows Phone, iOS, Android, BlackBerry και Symbian χρησιμοποιούνται για την υποστήριξη των μικρών αυτών έξυπνων κινητών συσκευών. Η νέα τάση στις τελευταίες είναι να περιλαμβάνουν όσον το δυνατό περισσότερες λειτουργίες ενός επιτραπέζιου Ηλεκτρονικού Υπολογιστή, με επιπρόσθετα ένα πλήθος από εργαλεία όπως αισθητήρες, GPS και φωτογραφική μηχανή.

Αυτή η αυξανόμενη διείσδυση των Τεχνολογιών της Πληροφορίας και Επικοινωνίας (ΤΠΕ) στην καθημερινή ζωή έχει ως συνέπεια να εμφανίζονται συσχετισμοί ανάμεσα σε γνωστικά πεδία τα οποία μέχρι τώρα φαίνονταν εντελώς άσχετα μεταξύ τους. Εισάγεται ο όρος *εκπαιδευτική τεχνολογία*, ο οποίος περιγράφει τη διαδικασία χρησιμοποίησης τεχνολογικών μέσων για

εκπαιδευτικούς σκοπούς, με κύριο αποτέλεσμα το σχεδιασμό και την υλοποίηση εκπαιδευτικού λογισμικού. Ως *εκπαιδευτικό λογισμικό* ορίζεται το προϊόν της τεχνολογίας με το οποίο προσπαθούμε να διδάξουμε ένα γνωστικό αντικείμενο υλοποιώντας συγκεκριμένη παιδαγωγική φιλοσοφία και συγκεκριμένη εκπαιδευτική στρατηγική. Η δημιουργία κατάλληλων συνθηκών για την προώθηση της αποτελεσματικής μάθησης αποτελεί τον απώτερο σκοπό της Διδακτικής των Επιστημών. Η εκπαιδευτική εμπειρία σε σχολεία πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, δείχνει ότι σημαντική μέθοδος για την πρόκληση του ενδιαφέροντος του μαθητή είναι η δημιουργία **κινήτρου**. Γι' αυτό, θεωρούμε ότι ένα λογισμικό σε περιβάλλον παιχνιδιού μπορεί να βοηθήσει στην απόκτηση και ανάπτυξη δεξιοτήτων αξιοποιώντας τις δυνατότητες που παρέχονται από τις σύγχρονες κινητές συσκευές.

1.1 Σκοπός και αντικείμενο έρευνας

Ο στόχος αυτής της διατριβής είναι ο σχεδιασμός, η υλοποίηση και η αξιολόγηση μιας δικτυακής εφαρμογής – εκπαιδευτικού παιχνιδιού σε πλατφόρμα κινητού υπολογισμού. Η εφαρμογή θα ρησιμοποιεί, είτε σύνδεση Bluetooth για τη δημιουργία τοπικού δικτύου, είτε Wi-Fi για την σύνδεση διαδικτυακά. Πρόκειται για ένα παιχνίδι τύπου κουίζ με εικονική περιήγηση των παικτών σε ένα κοινό λαβύρινθο και δυνατότητα συνεργασίας μεταξύ τους. Το θέμα προορίζεται κυρίως για παιδιά των τελευταίων δύο τάξεων του δημοτικού στο μάθημα «Πληροφορική και Νέες Τεχνολογίες». Οι μαθησιακοί στόχοι πρέπει να είναι συμβατοί με το πρόγραμμα σπουδών. Η εφαρμογή θα αναπτυχθεί για το λειτουργικό σύστημα android [01], καθώς είναι το ευρύτερα διαδεδομένο, πιο προσιτό και δωρεάν στη χρήση του. Η διεπαφή (GUI) θα είναι στην ελληνική γλώσσα.

Το αποτέλεσμα θα είναι η δημιουργία ενός ψυχαγωγικού συνεργατικού παιχνιδιού πληροφορικής με εκπαιδευτικό προσανατολισμό, αναδεικνύοντας τις δυνατότητες που προσφέρουν οι σύγχρονες κινητές πλατφόρμες. Σύμφωνα με τη Θεωρία για το Κίνητρο Επίτευξης (McClelland, 1953), οι άνθρωποι έχουν την τάση να πασχίζουν για την επιτυχία και να επιλέγουν δραστηριότητες με προσανατολισμό την επιτυχία/αποτυχία επίτευξης κάποιου στόχου. Το παιχνίδι χρησιμοποιείται ως κίνητρο για το χρήστη, ώστε αυτός να αποκτήσει συγκεκριμένες δεξιότητες και γνώσεις καθώς περιηγείται στο λογισμικό και ολοκληρώνει τις διαδικασίες του παιχνιδιού. Ιδιαίτερη έμφαση δίνεται στη συνεργατικότητα μεταξύ των χρηστών, καθώς σύμφωνα με τη συμβουλή του γνωστού Ρώσου φιλόσοφου και ψυχολόγου της εκπαίδευσης Vygotsky στις κοινωνικοπολιτισμικές θεωρίες, η μάθηση επιτυγχάνεται καλύτερα

μέσω συνεργατικών δραστηριοτήτων. Τα εκπαιδευτικά παιχνίδια δίνουν στους μαθητές τη δυνατότητα να διδαχθούν σε ένα πλαίσιο που αυξάνει την κινητοποίηση και τον ενθουσιασμό αλλά και την προσοχή στη μαθησιακή διαδικασία, κάνοντας το μάθημα πιο ευχάριστο.

1.1.1 Αναγκαιότητα και σπουδαιότητα έρευνας

Τα οφέλη της χρήσης εκπαιδευτικών λογισμικών στην εκπαίδευση, όπως έχει προταθεί από την B.E.C.T.A. (2001) είναι:

- Ανάπτυξη δεξιοτήτων στις Νέες Τεχνολογίες και Επικοινωνίες (παρόλο που αυτό μπορεί να είναι χρονοβόρο).
- Αυξημένο κίνητρο (παρόλο που τα παιχνίδια μπορεί να είναι περίπλοκα για τον πληθυσμό της τάξης).
- Ενθάρρυνση για συνεργατικότητα (αν και τα παιχνίδια μπορούν να είναι τόσο εθιστικά, που να χαθεί η εστίαση στο εκπαιδευτικό μέρος).
- Ανάπτυξη κριτικής ικανότητας.

Ο Prensky έχει αναγνωρίσει ένα συνδυασμό 12 στοιχείων, που κάνουν τα ηλεκτρονικά παιχνίδια ελκυστικά. Αυτά συνοψίζονται στον πίνακα 1.1.

Χαρακτηριστικά ενός ηλεκτρονικού παιχνιδιού	Πώς συνεισφέρει στην προσέλκυση του παίκτη
Διασκέδαση	Ψυχαγωγία και ευχαρίστηση
Παίξιμο	Έντονη και παθιασμένη εμπλοκή
Κανόνες	Δομή
Στόχοι	Κίνητρο
Αλληλεπίδραση	Να κάνει (πχ τη δραστηριότητα)
Αποτελέσματα και ανατροφοδότηση	Μάθηση
Προσαρμογή	Ροή

Χαρακτηριστικά ενός ηλεκτρονικού παιχνιδιού	Πώς συνεισφέρει στην προσέλκυση του παίκτη
Νίκη	Προσωπική ικανοποίηση
Διαμάχη/ συναγωνισμός/ πρόκληση και αντιπαλότητα	Αδρεναλίνη
Επίλυση προβλημάτων	Έναυσμα δημιουργικότητας
Συνεργασία	Κοινωνικές ομάδες
Αντιπροσώπευση και πλοκή	Συναίσθημα

Πίνακας 1.1: Ελκυστικά χαρακτηριστικά των ηλεκτρονικών παιχνιδιών [02].

Επιπλέον, η γλώσσα που χρησιμοποιείται για την επεξεργασία και παρουσίαση από το λογισμικό είναι σημαντικότερο χαρακτηριστικό, καθώς το να είναι στη μητρική γλώσσα του καθενός, το κάνει προσιτό σε περισσότερα άτομα, ακόμα και σε εκείνα που δεν έχουν αρκετές γνώσεις ξένων γλωσσών. Η υποστήριξη των ελληνικών σε εφαρμογές-παιχνίδια είναι πολύ περιορισμένη μέχρι στιγμής.

1.1.2 Τα μοντέλα των Gagne και Laurillard.

Η φορητότητα που χαρακτηρίζει τις κινητές πλατφόρμες, τις καθιστά ιδανικές για μάθηση από απόσταση. Παρακάτω δίδεται μια σύντομη περιγραφή για δύο από τα πιο δημοφιλή εκπαιδευτικά μοντέλα με εκτενή εφαρμογή στη μάθηση εξ αποστάσεως.

Ένα από τα πιο ευρέως χρησιμοποιούμενα σύνολα κατευθυντήριων γραμμών για το μαθησιακό σχεδιασμό πηγάζει από την εκπαιδευτική θεωρία του Robert Gagne. Σε αυτό το μοντέλο, η έμφαση δίνεται στην αύξηση της απόδοσης του μαθητή. Η γνωστική του θεωρία σκιαγραφεί εννέα διαδοχικά «Γεγονότα Διδασκαλίας» που παρέχουν τις απαραίτητες οδηγίες για τη μάθηση (Gagne, Briggs και Wager, 1992). Αυτές είναι:

1. Προσέλκυση της προσοχής.
2. Πληροφόρηση του μαθητή για τους στόχους του μαθήματος.
3. Παροχή ερεθισμάτων για την ανάκληση προαπαιτούμενων γνώσεων.

4. Παρουσίαση του υλικού προς μάθηση.
5. Παροχή οδηγιών μάθησης.
6. Πρόκληση των μαθησιακών ενεργειών.
7. Επανατροφοδότηση του μαθητή.
8. Εκτίμηση της απόδοσης των μαθησιακών ενεργειών.
9. Ενίσχυση της συγκράτησης και μεταφοράς της μάθησης.

Ο Gagne πίστευε στη «μαθησιακή ιεραρχία», μία δόμηση δεξιοτήτων κατά την οποία ο εκπαιδευτικός πρέπει πρώτα να αναγνωρίσει και να βεβαιωθεί ότι οι μαθητές κατέχουν τις προαπαιτούμενες δεξιότητες. Πολλά μαθήματα που κάνουν χρήση εκπαιδευτικών λογισμικών που βασίζονται στην εξάσκηση, στην προσομοίωση και κυρίως στην καθοδηγούμενη μάθηση (tutorials) έχουν σχεδιαστεί σύμφωνα με τις εκπαιδευτικές αρχές του Gagne, ή παρόμοια μοντέλα. Θεωρούσε ότι μόνο η καθοδηγούμενη μάθηση μπορεί να «σταθεί από μόνη της», δηλαδή να περαιώσει όλα τα γεγονότα διδασκαλίας, σε αντίθεση με τα άλλα είδη λογισμικού που απαιτούν δραστηριότητες με συνεχείς κατευθύνσεις από τον εκπαιδευτικό πριν, κατά τη διάρκεια και μετά τη χρήση του λογισμικού [14].

Η Laurillard (1993, 2002) προτείνει μια κάπως διαφορετική σκοπιά για τον εκπαιδευτικό σχεδιασμό με ενσωμάτωση της τεχνολογίας στη διδασκαλία. Στο βιβλίο της, η Laurillard (1993) [15] περιγράφει πέντε βήματα για το σχεδιασμό της διδασκαλίας που είναι απαραίτητα για να υπάρξει «βαθιά» μάθηση. Αυτά είναι:

1. Ο δάσκαλος προσφέρει καθοδήγηση στον μαθητή.
2. Ο μαθητής ανταποκρίνεται.
3. Ο δάσκαλος δέχεται ανατροφοδότηση από το μαθητή.
4. Ο μαθητής ανταποκρίνεται χρησιμοποιώντας την ανατροφοδότηση από το δάσκαλο.

5. Ο κύκλος αυτός επαναλαμβάνεται όσο ο δάσκαλος προσφέρει περαιτέρω καθοδήγηση [17].

Στο διαλογικό της πλαίσιο (conversational framework) το 2002, τονίζει την ιδιαίτερη σημασία του διαλόγου μεταξύ μαθητών και δασκάλων σε σχέση με την απλή διαβίβαση πληροφοριών από το δάσκαλο στο μαθητή [16]. Σε αντίθεση με τον Gagne, το μοντέλο Laurillard της διδασκαλίας σχεδιάστηκε έχοντας υπόψη τεχνολογία διαδραστικής φύσης.

Η μεθοδολογία που ακολουθείται για την ανάπτυξη των εκπαιδευτικών χαρακτηριστικών της εφαρμογής σε αυτήν την μεταπτυχιακή διατριβή βασίζεται κυρίως στα «Γεγονότα Διδασκαλίας» από το μοντέλο του Gagne.

1.1.3 Προσδοκώμενα αποτελέσματα

Με βάση τη θεωρία της πολλαπλής νοημοσύνης (Gardner, 1993), οι άνθρωποι έχουν ένα πλήθος ξεχωριστών ειδών ευφυΐας, που μπορεί να ποικίλουν σε ισχύ. Συνεπώς, η διδασκαλία επιστημονικών εννοιών χρησιμοποιώντας ποικιλία προσεγγίσεων, όπως οι νέες τεχνολογίες, θα οδηγήσει σε μεγαλύτερη επίτευξη μάθησης από όταν χρησιμοποιούνται μόνο γλωσσικές και μαθηματικές προσεγγίσεις. Επιπρόσθετο βοήθημα από τη χρήση υπολογιστικών συστημάτων στην εκπαίδευση είναι η αύξηση της ικανότητας των μαθητών για λογικούς συμπερασμούς. Από το συγκεκριμένο λογισμικό προσδοκάται:

- Να είναι εύκολο στη χρήση.
- Να ψυχαγωγήσει και να ταυτόχρονα να μεταφέρει γνώσεις.
- Να χρησιμοποιεί γλώσσα απλή και κατανοητή.
- Να μην κάνει καταχρήσεις των χρησιμοποιούμενων δυνατοτήτων.
- Να δίνει στο χρήστη πάντα μια πόρτα διαφυγής.
- Να προωθήσει τη συνεργατική αλλά και την εξατομικευμένη μάθηση.

1.1.4 Βασικά ερευνητικά ερωτήματα:

Το λογισμικό που θα αναπτυχθεί θα πρέπει να είναι απλό, φορητό, ευέλικτο, συντηρήσιμο και επαναχρησιμοποιήσιμο. Θα πρέπει να εξεταστούν ερωτήματα όπως τα ακόλουθα:

- Κατά πόσο η χρήση ενός ψυχαγωγικού συνεργατικού παιχνιδιού με εκπαιδευτικό προσανατολισμό μπορεί να βοηθήσει τη διαδικασία μάθησης και την ανάπτυξη δεξιοτήτων;
- Ποιοι μπορούν να παίξουν το συγκεκριμένο παιχνίδι και σε ποιους απευθύνεται;
- Τι μπορεί ακριβώς να κάνει το εν λόγω λογισμικό;
- Πως μπορεί να υλοποιηθεί;
- Ποιες δυνατότητες των σύγχρονων πλατφορμών κινητού υπολογισμού μπορούν να αξιοποιηθούν και με ποιο τρόπο;
- Ποια είναι η ψυχαγωγική και ποια η διδακτική απόδοση;
- Ποιοι θα αποτελούν τους αξιολογητές του λογισμικού και με ποιο τρόπο θα επιλεγούν οι αξιολογητές;

Η ανάλυση, η σχεδίαση και η υλοποίηση της εφαρμογής θα γίνει σύμφωνα με τις αρχές της αντικειμενοστρεφούς τεχνολογίας, ενώ στη συνέχεια θα αξιολογηθούν τα επιμέρους χαρακτηριστικά και αποτελέσματα από τη χρήση της μέσω ερωτηματολογίων που θα δοθούν σε έναν αριθμό χρηστών.

Το μορφωτικό παιχνίδι είναι ένα δύσκολο είδος λογισμικού για αξιολόγηση, δεδομένου ότι είναι αμφίβολο αν οι ικανότητες που προωθούνται από το παιχνίδι είναι δυνατόν να μεταφέρουν και έννοιες. Αν η αξιολόγηση του εκπαιδευτικού λογισμικού οριστεί γενικά ως η συλλογή, η ανάλυση και η ερμηνεία δεδομένων που αποσκοπούν στην εκτίμηση της αποτελεσματικότητας του και της αποδοτικότητάς του (Χ. Παναγιωτακόπουλος, 2003), τότε θα πρέπει αρχικά να προσδιοριστεί ένα γενικότερο πλαίσιο για την πραγματοποίηση μιας συστηματικής αξιολόγησης.

1.1.5 Συμπεράσματα

Αυτό που απαιτείται είναι ένα εργαλείο με έμφαση στον εύχρηστο τρόπο παρουσίασης της πληροφορίας, ώστε το παιδί να αισθάνεται ευχαρίστηση με αυτό που επιτυγχάνει. Με άλλα λόγια να του δίνει ένα κίνητρο όπως ένα παιχνίδι. Αυτό επιδιώκεται να επιτευχθεί με τη συνεργασία και τη χρήση των πολλαπλών τρόπων εισαγωγής του ερωτήματος. Η βάση γνώσης θα πρέπει να μην περιλαμβάνει προχωρημένες έννοιες, αλλά προσιτές και κατανοητές για τις ηλικίες στις οποίες απευθύνεται. Τέλος, κρίσιμος παράγοντας είναι η γλώσσα που χρησιμοποιείται για την επεξεργασία και παρουσίαση από το λογισμικό, καθώς επιβάλλεται να είναι στη μητρική γλώσσα του χρήστη, το οποίο στην περίπτωσή μας είναι τα ελληνικά.

1.2 Ερευνητική πρωτοβουλία

Στην κατηγορία των εκπαιδευτικών λογισμικών συμπεριλαμβάνονται παιχνίδια, για κάθε είδους λειτουργικό σύστημα, τα οποία εκτός από ψυχαγωγικό χαρακτήρα συμβάλλουν στη μάθηση, την ανάπτυξη δεξιοτήτων, ενώ όταν τα παιχνίδια αυτά υποστηρίζουν πολλαπλούς παίκτες βοηθούν στην ανάπτυξη της συνεργατικότητας.

1.2.1 Λειτουργικά συστήματα

Το Android είναι ένα λειτουργικό σύστημα ανοικτού κώδικα με ιδιόκτητα μέρη, για πλατφόρμες κινητού υπολογισμού, που εισήχθη σχετικά πρόσφατα, το 2008 από τη Google. Είναι το δημοφιλέστερο αυτήν τη στιγμή, αφού έχει επιδείξει μια εντυπωσιακή ανοδική πορεία, το οποίο καθιστά σπουδαιότερη την ανάπτυξη ενός δικτυακού παιχνιδιού πάνω σε αυτό, καθώς και την αξιολόγησή του. Από την απέναντι πλευρά το Symbian, που ήταν το δημοφιλέστερο τη δεκαετία του 2000, χρησιμοποιείται όλο και λιγότερο, αφού ακόμα και ο μεγαλύτερος υποστηρικτής του, η εταιρεία κατασκευής κινητών συσκευών Nokia, αποφάσισε να στραφεί προς το Windows Phone της Microsoft. Το τελευταίο αν και όχι ιδιαίτερα δημοφιλές, κατέχει σταθερό μερίδιο στην αγορά.

OS Περίοδος	Android	iOS	Windows Phone	BlackBerry OS	Symbian
2ο Τρίμηνο 2014	84,7 %	11,7 %	2,5 %	0,5 %	0,0 %
1ο Τρίμηνο 2014	83,6 %	15,6 %	2,7 %	0,5 %	0,0 %
1ο Τρίμηνο 2013	75,0 %	17,3 %	3,2 %	2,9 %	0,6 %
1ο Τρίμηνο 2012	59,0 %	23,1 %	2,2 %	6,4 %	6,8 %
1ο Τρίμηνο 2011	36,1 %	18,3 %	2,6 %	13,6 %	26,0 %

Πίνακας 1.2: Παγκόσμιο μερίδιο αγοράς έξυπνων κινητών τηλεφώνων ανά λειτουργικό σύστημα (Πηγή IDC: World-Wide Smartphone Shipments).

Το λειτουργικό σύστημα Android, ενώ το 2009 είχε ξεκινήσει με μερίδιο αγοράς της τάξης του 1%, έναντι 10% του iOS και 50% του Symbian [03], σήμερα έχει επικρατήσει στην αγορά με μερίδιο της τάξης του 85% και μοναδικό ουσιαστικά ανταγωνιστή του το iOS, αλλά με μεγάλη διαφορά. Το iOS είναι ένα λειτουργικό σύστημα κλειστού κώδικα, αποκλειστικά για τις κινητές πλατφόρμες της εταιρείας Apple. Αντίθετα από το Windows Phone της Microsoft και το Android της Google, η Apple δεν δίνει την άδεια για την εγκατάσταση του λογισμικού iOS σε συσκευές που δεν είναι κατασκευής Apple.

1.2.2 Δικτυακά παιχνίδια

Ενώ έχει ήδη αναπτυχθεί μια πληθώρα εφαρμογών για τις «έξυπνες» πλατφόρμες κινητού υπολογισμού, υπάρχει σημαντική έλλειψη λογισμικών με εκπαιδευτικό χαρακτήρα και κυρίως με υποστήριξη πολλών παικτών ταυτόχρονα (multiplayer) από διαφορετικές κινητές πλατφόρμες σε τοπικό επίπεδο. Η δυνατότητα για τοπικά multiplayer παιχνίδια είναι περιορισμένης αξιοποίησης στον τομέα των smartphones. Τα περισσότερα παιχνίδια που διαφημίζονται σαν multiplayer είναι ουσιαστικά διαδικτυακά (μέσω internet) [04], ενώ ειδικά αυτά που εκμεταλλεύονται την τεχνολογία bluetooth ακόμα λιγότερα. Επιπλέον, η πλειοψηφία των διαθέσιμων εφαρμογών δεν είναι στην ελληνική γλώσσα.

Παρακάτω, παρατίθενται συνοπτικά ενδεικτικά παιχνίδια σε πλατφόρμα Android, ορισμένα από τα οποία διαθέτουν εκπαιδευτικό περιεχόμενο, δυνατότητα multiplayer ή και υποστήριξης ελληνικής γλώσσας.

- Το «SCRABBLE» [05] είναι ένα ψυχαγωγικό, εκπαιδευτικό παιχνίδι λέξεων, που υποστηρίζει πολλαπλούς παίκτες μέσω internet, τυχαίους ή μη, αλλά και συνομιλία σε πραγματικό χρόνο (chat). Στα εκπαιδευτικά του χαρακτηριστικά ξεχωρίζει η δυνατότητα που επιτρέπει στους παίκτες να δουν την καλύτερη δυνατή επιλογή που θα μπορούσαν να έχουν κάνει στην προηγούμενή τους κίνηση. Στα αρνητικά η έλλειψη υποστήριξης τοπικού multiplayer και ελληνικής γλώσσας.



Εικόνα 1.1: SCRABBLE.

- Το «New Words with Friends» [06] είναι κι αυτό ένα παιχνίδι με σχεδόν πανομοιότυπα χαρακτηριστικά με το προηγούμενο.



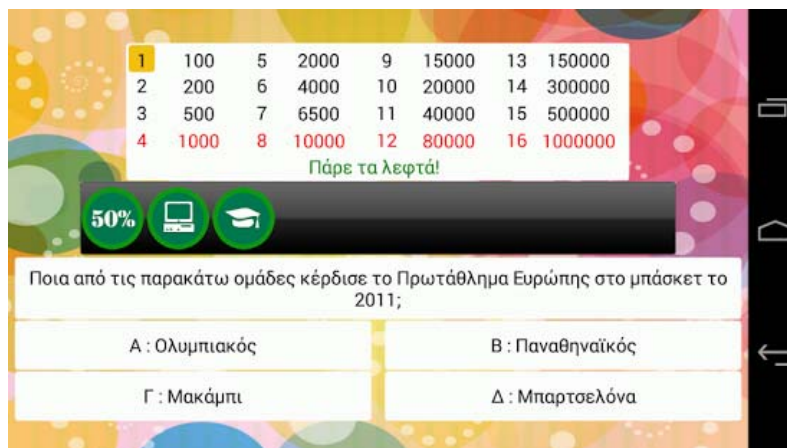
Εικόνα 1.2: New Words with Friends.

- Το «Guerrilla Bob» [07] είναι ένα διασυστημικό (cross-platform) multiplayer συνεργατικό παιχνίδι, αλλά δεν έχει εκπαιδευτικά στοιχεία, ενώ κι αυτό δεν υποστηρίζει τοπικό multiplayer. Απαιτεί συσκευές υψηλής απόδοσης (high-end).



Εικόνα 1.3: Guerilla Bob.

- Το «Γίνε Πλούσιος» [08] είναι ένα κουίζ γνώσεων στα ελληνικά το οποίο είναι μόνο για έναν παίκτη.



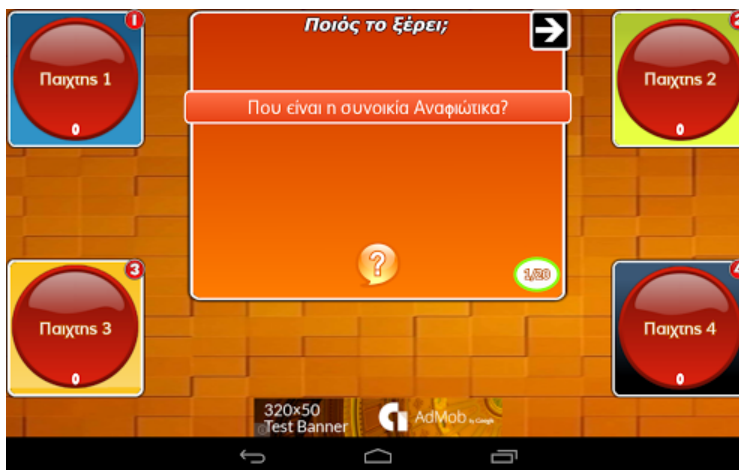
Εικόνα 1.4: Γίνε Πλούσιος.

- Το «4 Player Reactor» [09] υποστηρίζει multiplayer παιχνίδι πάνω σε μία μόνο συσκευή. Έχει κάποια εκπαιδευτικά στοιχεία, μέσω παιχνιδιών γνώσεων και σπαζοκεφαλίες και μάλιστα στα ελληνικά, αλλά δεν προωθεί τη συνεργασία μεταξύ παικτών αφού είναι ανταγωνιστικό. Ωστόσο το multiplayer περιορίζεται χωροταξικά, καθώς δε χρησιμοποιούνται οι τεχνολογίες ασύρματης ζεύξης. Επιπρόσθετα, η τελευταία έκδοση απαιτεί ένα μικρό αντίτιμο για τη διάθεση.



Εικόνα 1.5: 4 Player Reactor.

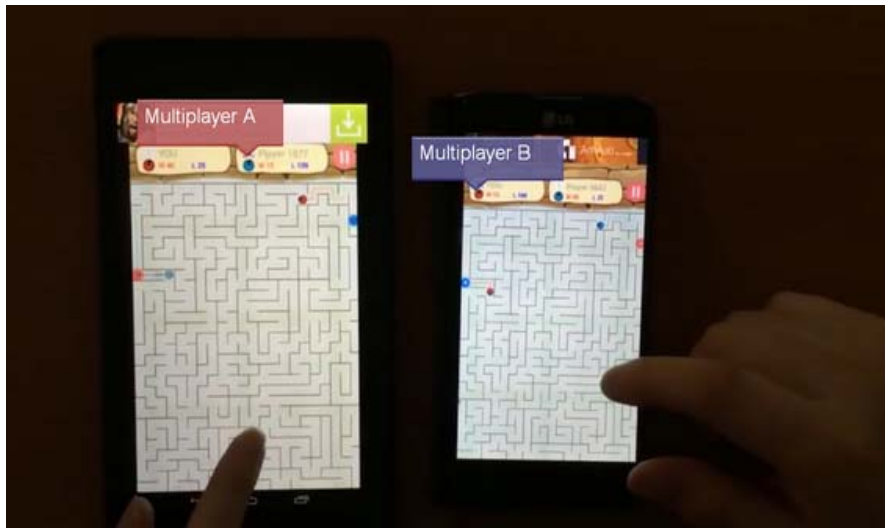
- Το «Ποιος το ξέρει;» [10] είναι ένα ακόμη παιχνίδι τύπου κουίζ, το οποίο κυκλοφόρησε πρόσφατα, είναι ελεύθερο και υποστηρίζει 2 έως 4 παίκτες. Είναι στην ελληνική γλώσσα και έχει όμοια χαρακτηριστικά με το «4 Player Reactor» που προαναφέρθηκε. Παίζεται κι αυτό σε μία συσκευή από όλους τους παίκτες.



Εικόνα 1.6: Ποιος το ξέρει;

- «Maze King» (Ο βασιλιάς του λαβυρίνθου) [11]. Είναι ένα παιχνίδι λαβυρίνθου για 1 ή 2 παίκτες από όλον τον κόσμο. Η λειτουργία δύο παικτών αφορά το συναγωνισμό μεταξύ τους και η σύνδεση επιτυγχάνεται μέσω των λογαριασμών στη Google που απαιτούνται να διαθέτουν και οι δύο. Η κίνηση της μπίλιας πραγματοποιείται με την κίνηση του

δακτύλου προς την αντίστοιχη κατεύθυνση στην οθόνη αφής. Επιτρέπεται η καταγραφή βαθμολογιών και η κατάταξη του καθενός μόνο στους διακομιστές της Google.



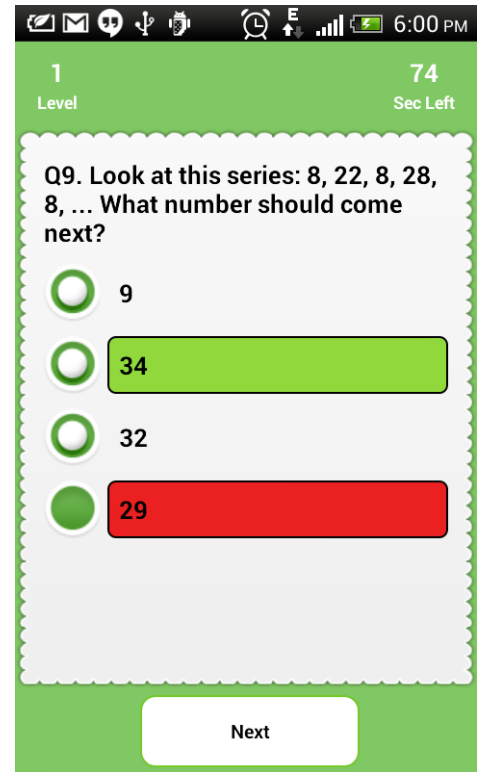
Εικόνα 1.7: Maze King.

- Το «Tic Tac Toe King» [12] είναι το κλασικό παιχνίδι τρίλιζας όπου ο παίκτης μπορεί να κερδίσει όταν τοποθετεί τουλάχιστο τρία ίδια στοιχεία («O» ή «X») σε διαγώνια, οριζόντια ή κάθετη σειρά. Όπως και το «Maze King», υποστηρίζει 1 ή 2 παίκτες από όλον το κόσμο μέσω των λογαριασμών τους στη Google.



Εικόνα 1.8: Tic Tac Toe King.

- Το «IT Logic Quiz» [13] είναι ένα ψυχαγωγικό παιχνίδι όπου ο παίκτης δοκιμάζει τις γνώσεις και δεξιότητές του στη λογική της Τεχνολογίας της Πληροφορικής. Είναι ένα εκπαιδευτικό κουίζ λογικής για έναν παίκτη, που απευθύνεται σε παιδιά και εφήβους.



Εικόνα 1.9: IT Logic Quiz.

Εκτός από τα παραπάνω παιχνίδια που αναφέρονται ενδεικτικά, έχουν αναπτυχθεί αρκετά ακόμη παρόμοια ή παραλλαγές αυτών με όμοια χαρακτηριστικά. Στα πλαίσια της παρούσας διατριβής θα αναπτυχθεί παιχνίδι με όνομα «Λαβύρινθος».

1.2.3 Χαρακτηριστικά Εφαρμογών/Παιχνιδιών σε κινητές πλατφόρμες με Android.

Όνομα Εργαλείου	Ένας παίκτης (single-player)	Πολλαπλοί παίκτες μέσω διαδικτύου	Πολλαπλοί παίκτες τοπικά	Συνεργατικό multi-player	Εκπαιδευτικά στοιχεία	Λαβύρινθος	Συνομιλία (chat)	Διασυστημικό (cross-platform) παιχνίδι	Υποστήριξη Ελληνικής γλώσσας	Ελεύθερο
SCRABBLE	*	*			*		*	*		*
New Words with Friends	*	*			*		*	*		*
Guerrilla Bob	*	*		*				*		Έκδοση LITE
Γίνε Πλούσιος	*				*				*	*
4 Player Reactor			*		*				*	
Ποιος το ξέρει;			*		*				*	*
Maze King	*	*				*		*		*
Tic Tac Toe King	*	*						*		*
IT Logic Quiz	*				*					*
Λαβύρινθος	*	*	*	*	*	*			*	*

Πίνακας 1.3: Συγκριτικός πίνακας με τα βασικά χαρακτηριστικά ενδεικτικών παιχνιδιών για κινητές πλατφόρμες με λειτουργικό σύστημα Android (συμπεριλαμβάνεται το παιχνίδι «Λαβύρινθος» που θα αναπτυχθεί στο πλαίσιο της παρούσας διατριβής).

Κεφάλαιο 2

Ανάλυση και Σχεδιασμός

Στην παρούσα διατριβή πρόκειται να σχεδιαστεί, να υλοποιηθεί και να αξιολογηθεί ο «Λαβύρινθος», μια δικτυακή εφαρμογή-παιχνίδι, σε πλατφόρμα κινητού υπολογισμού με λειτουργικό σύστημα android, εστιάζοντας στον ψυχαγωγικό και εκπαιδευτικό χαρακτήρα, προσπαθώντας παράλληλα να αξιοποιήσει τις δυνατότητες που παρέχουν οι σύγχρονες πλατφόρμες (όπως οι αισθητήρες). Το παιχνίδι θα παίζεται από έναν ή και περισσότερους παίκτες, μέσω ασύρματης σύνδεσης, είτε τοπικά με Bluetooth, είτε διαδικτυακά με Wi-Fi. Η διεπαφή θα παρέχει υποστήριξη της ελληνικής γλώσσας.

2.1 Μεθοδολογία

Αρχικά, πρόκειται να γίνει μια ενδελεχής ανάλυση των ζητούμενων με βάση τις αρχές της αντικειμενοστρεφούς τεχνολογίας. Για τη σχεδίαση θα χρησιμοποιηθεί η σημειογραφία UML, καθώς είναι διεθνές standard. Η υλοποίηση της εφαρμογής θα γίνει σε γλώσσα Java για Android.

Συνοπτικά τα στάδια της εργασίας είναι τα εξής:

1. Συγκέντρωση των ζητούμενων στοιχείων του προβλήματος

2. Ανάλυση
3. Σχεδιασμός σε UML
4. Υλοποίηση σε Java για Android
5. Αξιολόγηση

Τα παραπάνω στάδια δεν είναι ο ευκολότερος ή συντομότερος τρόπος επίλυσης, αλλά η χρήση και κατανόησή τους θα βοηθήσει αργότερα σε γρηγορότερη, ευκολότερη και καλύτερη επίλυση μεγαλύτερων και σημαντικότερων, πραγματικών προβλημάτων.

Το τελικό αποτέλεσμα θα είναι ένα λογισμικό απλό, φορητό, ευέλικτο, συντηρήσιμο και επαναχρησιμοποιήσιμο.

2.2 Εκπαιδευτικά χαρακτηριστικά

Τα εκπαιδευτικά χαρακτηριστικά του λογισμικού «Λαβύρινθος» θα αξιοποιούν κάποια στοιχεία του conversational framework της Laurillard, αλλά κυρίως το καθοδηγητικό μοντέλο του Gagne, που στηρίζεται στις αρχές των συμπεριφοριστικών προσεγγίσεων και επεξεργασίας της πληροφορίας. Συγκεκριμένα, τα λογισμικά «κλειστού τύπου» πρακτικής και εξάσκησης (drill and practice) των συμπεριφοριστικών μεθόδων είναι κατάλληλα, εκτός από την εξάσκηση και εμπέδωση της γνώσης, και για την αξιολόγηση των μαθητών, χρησιμοποιούμενα σαν τεστ γνώσεων. Οι στρατηγικές αυτές δίνουν εύκολα παρατηρήσιμες ενδείξεις επάρκειας δεξιοτήτων, αλλά και αντικειμενικές μετρήσεις επίτευξης αποτελεσμάτων, ενώ χρησιμοποιούνται ευρέως και σήμερα. Επιπλέον των παραπάνω, το δικτυακό κομμάτι θα προσφέρει τη δυνατότητα για παιχνίδι συνεργασίας, προσθέτοντας έτσι κοινωνικοπολιτισμικά χαρακτηριστικά.

Εκπαιδευτικό Χαρακτηριστικό	Μοντέλο Laurillard	Μοντέλο Gagne	Λαβύρινθος
Εισήγηση - Επίδειξη	✓	✓	-
Περιγραφή του μαθησιακού υλικού από την άποψη του εκπαιδευτικού	✓	✓	✓
Παρακίνηση του μαθητή κάνοντας σύνδεση με τα προηγούμενα.	✓	✓	✓
Πρακτική και εξάσκηση (ασκήσεις)	✓	✓	✓

Εκπαιδευτικό Χαρακτηριστικό	Μοντέλο Laurillard	Μοντέλο Gagne	Λαβύρινθος
Άμεση αποσαφήνιση των εννοιών	✓	-	-
Αλληλεπίδραση με το περιβάλλον	✓	✓	✓
Διόρθωση λαθών	-	✓	-
Ανατροφοδότηση ως μέρος της εκπαιδευτικής διαδικασίας	✓	-	✓
Κοινωνικοπολιτισμική μάθηση	-	-	✓
Εξακρίβωση αξιοπιστίας αξιολόγησης με ποικιλία από τεστ	-	✓	✓
Αξιολόγηση από την άποψη των μαθητών - Σωστή καθοδήγηση με αναδιαμόρφωση εικόνας	✓	-	-
Γεγονότα διδασκαλίας με αρχή και τέλος	-	✓	✓
Διδασκαλία με κύκλο γεγονότων	✓	-	-

Πίνακας 2.1: Σύγκριση και Αξιοποίηση Διδακτικών Μοντέλων στην Εφαρμογή Λαβύρινθος.

Το ερώτημα που τίθεται είναι κατά πόσο η χρήση ενός ψυχαγωγικού συνεργατικού παιχνιδιού με εκπαιδευτικό προσανατολισμό μπορεί να βοηθήσει τη διαδικασία μάθησης και την ανάπτυξη δεξιοτήτων.

2.3 Απαιτήσεις (Συγκέντρωση των ζητούμενων στοιχείων του προβλήματος)

Στον πραγματικό κόσμο, οι απαιτήσεις και τα ζητούμενα του προβλήματος συγκεντρώνονται με μια σειρά συνεντεύξεων, των αναλυτών προς αυτόν που ζητάει την επίλυση (πελάτη). Στη συγκεκριμένη περίπτωση απλά παρατίθεται μια λεπτομερής περιγραφή του προβλήματος, η περιγραφή του «Λαβύρινθου». Από αυτό το κείμενο θα εντοπιστούν τα ζητούμενα του προβλήματος.

ΛΑΒΥΡΙΝΘΟΣ

Το παιχνίδι μπορεί να υποστηρίξει έναν ή περισσότερους παίκτες ταυτόχρονα (multiplayer), το οποίο μπορεί να επιτυγχάνεται μέσω ασύρματης σύνδεσης τοπικά με Bluetooth ή Wi-Fi. Για να

επιτευχθεί η σύνδεση Wi-Fi, οι έξυπνες κινητές συσκευές θα παίζουν το ρόλο του πελάτη, που θα συνδέονται διαδικτυακά σε έναν εξυπηρετητή. Η μετακίνηση στο λαβύρινθο γίνεται με ένα σύστημα μέσω της οθόνης αφής (touchscreen), το οποίο απεικονίζεται στην οθόνη.

Η ακολουθία της διεκπεραίωσης ενός παιχνιδιού είναι η εξής:

1. Οι κινητές πλατφόρμες συνδέονται σε δίκτυο.
2. Το πρόγραμμα εντοπίζει την υποστηριζόμενη ανάλυση της οθόνης της συσκευής και ρυθμίζει αναλόγως τις παραμέτρους των στοιχείων που πρόκειται να απεικονιστούν, ώστε να χωρέσουν στην οθόνη.
3. Στις οθόνες των κινητών συσκευών δημιουργείται με γραφικά ένας λαβύρινθος με πολλές διαδρομές, που δημιουργούνται με τυχαίο τρόπο. Ο λαβύρινθος είναι κοινός για όλους. Για κάθε παίκτη υπάρχει μία διαφορετική είσοδος στο λαβύρινθο, όπου τοποθετείται και είναι κοντά στις εισόδους των άλλων παικτών, αλλά μόνο μία έξοδος για όλους. Οι εισοδοί και οι εξοδοί δημιουργούνται σε τυχαίες τοποθεσίες.
4. Υπάρχουν πολλές εναλλακτικές διαδρομές, και σε αυτές εμφανίζονται κάποιες Σφίγγες με αινίγματα και διαφορετική εμφάνιση, ανάλογα με την ερώτηση που κάνουν.
5. Ο κάθε παίχτης επιλέγει διαδρομή και πέφτει κάθε τόσο πάνω σε μία Σφίγγα, όπου θα πρέπει να λύσει το αίνιγμά της για να μπορέσει να περάσει από εκεί. Αν δύο παίχτες καταφέρουν να συναντηθούν, θα μπορούν να ακολουθήσουν κοινή διαδρομή και να απαντάνε από κοινού στις ερωτήσεις από εκεί και πέρα. Αν κάποιος δεν είναι σίγουρος για την απάντηση περνάει την ερώτηση στον άλλο.
6. Οι ερωτήσεις ενός αινίγματος μπορούν να είναι τύπου πολλαπλής-επιλογής με διαφορετικά επίπεδα δυσκολίας και κατηγορίας (Λογισμικό, Υλικό, Λειτουργικά Συστήματα, Δυαδικό Σύστημα, Ερώτηση Κρίσης). Ένα άλλο είδος αινίγματος είναι η εμφάνιση πίνακα με πολλές σωστές απαντήσεις και επιλογή των σωστών (π.χ. από τις 9 απαντήσεις να επιλεγούν οι 4 σωστές).
7. Ο κάθε παίχτης έχει μέχρι 4 προσπάθειες για να απαντήσει σωστά. Αν οι 2 παίχτες συναντηθούν, οι προσπάθειες θα αθροίζονται και θα είναι κοινές. Κατά το μήκος της διαδρομής, μπορούν να ανακαλυφθούν επιπλέον προσπάθειες σε διάφορες τοποθεσίες.

8. Από τις σωστές απαντήσεις του καθενός, καταγράφονται κοινοί πόντοι για όλους, ο οποίος αναγράφονται στην οθόνη.
9. Όλοι οι παίκτες πρέπει να βρουν την έξοδο από το λαβύρινθο. Στην επιτυχή αυτήν περίπτωση, ο κάθε παίκτης εισέρχεται στη Bonus Φάση. Έπειτα συνεχίζουν σε επόμενο παραγόμενο λαβύρινθο αυξημένου επιπέδου δυσκολίας.
10. Το παιχνίδι τελειώνει αν οι παίκτες καταφέρουν να βρουν την έξοδο από 2 συνολικά λαβύρινθους.

Είδος αιγιμάτων:

- 1ο Είδος. Δίνεται μια τυχαία ερώτηση και με τη μορφή πίνακα 9 απαντήσεις, εκ των οποίων οι 4 μόνο είναι σωστές. Ο χρήστης καλείται να βρει όλες τις σωστές απαντήσεις, έχοντας 5 προσπάθειες. Αν τις βρει όλες χωρίς λάθος κερδίζει μία προσπάθεια για τις επόμενες ερωτήσεις.
- 2ο Είδος. Εδώ δίνονται 5 ερωτήσεις. Στην κάθε ερώτηση προβάλλονται 4 απαντήσεις, εκ των οποίων μόνο μία είναι σωστή. Ο χρήστης καλείται να απαντήσει στις 4 από τις 5 ερωτήσεις σωστά. Οι ερωταπαντήσεις είναι κατηγοριοποιημένες και προβάλλονται με τη μορφή multiple-choice.

Το πρόγραμμα περιλαμβάνει επίσης μία βάση δεδομένων με ερωτήσεις και τις απαντήσεις σε αυτές.

Bonus Φάση: Προβάλλονται 5 κουτάκια. Το πρώτο κουτάκι δεν περιέχει τίποτα. Το δεύτερο περιέχει μία τυχαία ερώτηση τύπου Multiple-Choice με δικαίωμα επιλογής κατηγορίας. Αν ο χρήστης απαντήσει σωστά κερδίζει μια προσπάθεια, αλλιώς χάνει μία. Το τρίτο κουτάκι περιέχει 2 ερωτήσεις τύπου Multiple-Choice. Με καμία σωστή απάντηση χάνει μια προσπάθεια, με μία σωστή δεν κερδίζει/χάνει τίποτα, ενώ με δύο κερδίζει δύο προσπάθειες. Το τέταρτο κουτάκι περιέχει μία τυχαία ερώτηση τύπου Πίνακα. Αν ο χρήστης απαντήσει σωστά κερδίζει δύο προσπάθειες, αλλιώς χάνει μία. Το πέμπτο περιέχει ένα ζάρι. Με τυχαίο ρίξιμο του ζαριού ο χρήστης μπορεί να χάσει 1 προσπάθεια (αριθμός ζαριού 1 ή 2), να κερδίσει 1 προσπάθεια (αριθμός ζαριού 5), 2 προσπάθειες (αριθμός ζαριού 6), ή και να μην κερδίσει/χάσει τίποτα (αριθμός ζαριού 3 ή 4). Όπως και να 'χει, ο χρήστης συνεχίζει στον επόμενο λαβύρινθο.

2.4 Ανάλυση

Το παιχνίδι είναι συνεργατικό και όσο καλύτερα συνεργάζονται οι παίκτες, τόσο ευκολότερα θα προχωράνε παρακάτω. Ο κάθε παίκτης θα πρέπει να προσπαθήσει να βρει τον άλλο μέσα στον λαβύρινθο. Η συνεργατικότητα προκύπτει από τη στιγμή που θα συναντηθούν, όποτε και θα συνεχίσουν από κοινού τη διαδικασία εξόδου από αυτόν, επιλέγοντας μαζί τις διαθέσιμες διαδρομές και συνδυάζοντας τη δυναμική των γνώσεών τους να προσπερνάνε τα εμπόδια, απαντώντας στα προβλήματα και τις ερωτήσεις που τους υποβάλλονται.

Αφού κατανοηθούν τα ζητούμενα και το πρόβλημα, αναλύοντας τα δεδομένα στοιχεία θα αρχίσει να χτίζεται το μοντέλο, το οποίο θα είναι αντικειμενοστρεφές. Εδώ θα βοηθούσε ο σχεδιασμός ενός UML διαγράμματος δραστηριοτήτων, βασιζόμενο στην άνω περιγραφή του «Λαβυρίνθου». Θα βοηθούσε τους έχοντες και μη σχέση με το αντικείμενο, να καταλάβουν, να έχουν μια πιο σφαιρική άποψη της περιοχής του προβλήματος. Επίσης, ένα διάγραμμα περιπτώσεων χρήσεων θα έδειχνε στους μέλλοντικούς χρήστες, πως ο/οι αναλυτής/ές έχουν αντιληφθεί την λειτουργία του συστήματος. Στη συγκεκριμένη περίπτωση, θα συμπεριληφθούν με τη σειρά στην ενότητα του σχεδιασμού σε UML μαζί με τα υπόλοιπα διαγράμματα, για λόγους ταξινόμησης.

Γενικά, μια καλή σειρά (όχι αυστηρά επιβεβλημένη) με την οποία μπορεί να γίνει η μελέτη και σχεδιασμός των διαγραμμάτων, ώστε να επιτευχθεί η μέγιστη δυνατή απόδοση είναι η εξής:

1. Διαγράμματα Περιπτώσεων Χρήσεων – Καταστάσεων – Δραστηριοτήτων
2. Διαγράμματα Κλάσεων – Αντικειμένων
3. Διαγράμματα Πακέτων
4. Διάγραμμα Συστατικών
5. Διάγραμμα Ανάπτυξης

Προς το παρόν θα πρέπει να αναγνωριστούν και να βρεθούν οι πιθανές κλάσεις, οι οποίες θα χρησιμοποιηθούν αργότερα στο UML διάγραμμα κλάσεων ή και σε άλλα διαγράμματα. Αυτό

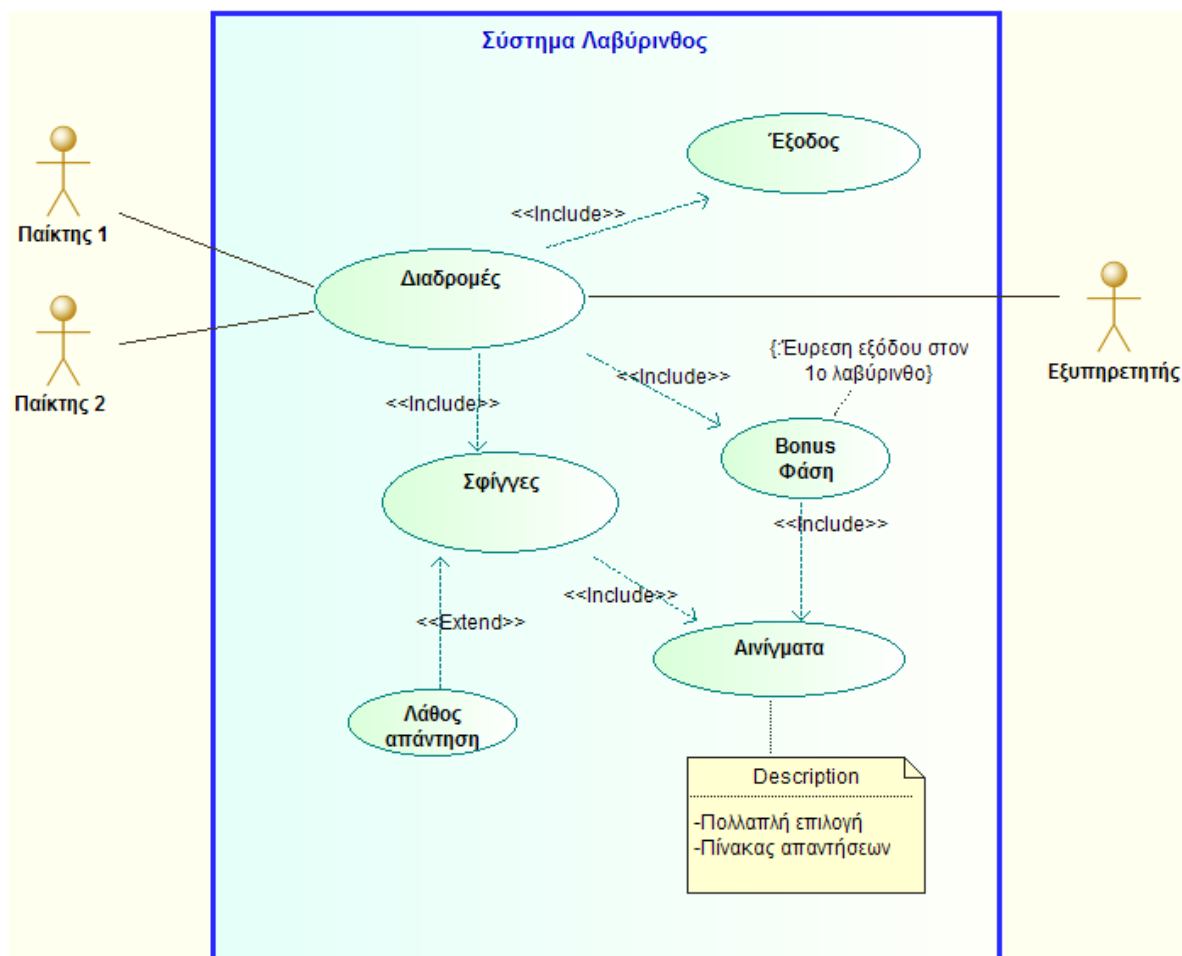
μπορεί να γίνει αναλύοντας το παραπάνω κείμενο της περιγραφής με τα ζητούμενα, σύμφωνα με παραδοσιακές τεχνικές, όπως η ανάλυση ουσιαστικών-ρημάτων (Abbot, 1983) [28, 29].

Μέρος λόγου	Συστατικό του μοντέλου	Παραδείγματα
Proper noun	αντικείμενο (object)	Πέτρος, Ειρήνη
Common noun	κλάση (class)	Χρήστης, παιχνίδι
Doing verb	μέθοδος (method)	δημιουργεί, επιλέγει, απαντάει
Being verb	κληρονομικότητα (inheritance)	είναι, είναι είδος του
Having verb	συσσωμάτωση (aggregation)	έχει, αποτελείται από, περιλαμβάνει
Modal verb	περιορισμός (constraint)	πρέπει να
Adjective	ιδιότητα (attribute)	σωστό, λάθος

Πίνακας 2.2: Ευρετική ανάλυση του Abbot για την αντιστοίχιση μερών του λόγου σε συστατικά μοντέλου.

2.4.1 Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)

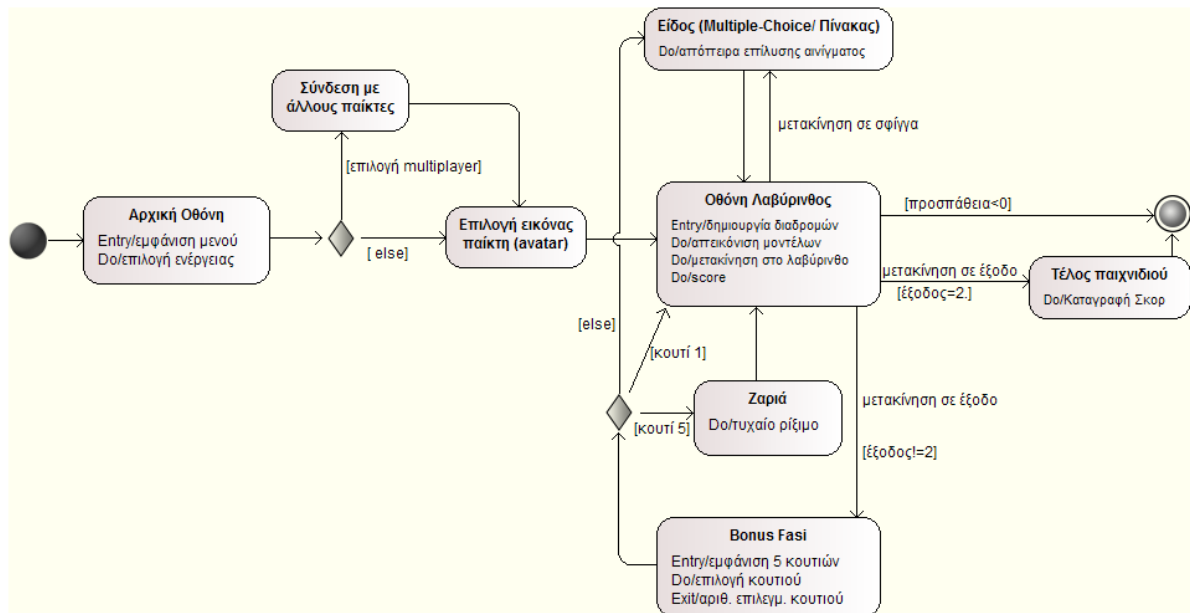
Οι περιπτώσεις χρήσης είναι η περιγραφή της συμπεριφοράς του συστήματος από την άποψη του χρήστη.



Σχήμα 2.1: UML Διάγραμμα Περιπτώσεων Χρήσης.

2.4.2 Διάγραμμα Καταστάσεων (Statechart Diagram)

Το διάγραμμα καταστάσεων περιγράφει τη δυναμική συμπεριφορά του συστήματος. Δηλώνονται οι καταστάσεις στις οποίες μεταβαίνει ένα σύστημα ή κάποιο αντικείμενό του, κατά τη διάρκεια της ζωής του. Η μετάβαση από τη μία κατάσταση σε μια άλλη προκαλείται από εξωτερικά ή εσωτερικά συμβάντα. Οι καταστάσεις από τις οποίες περνάει ο Λαβύρινθος απεικονίζονται στο παρακάτω UML Διάγραμμα Καταστάσεων.

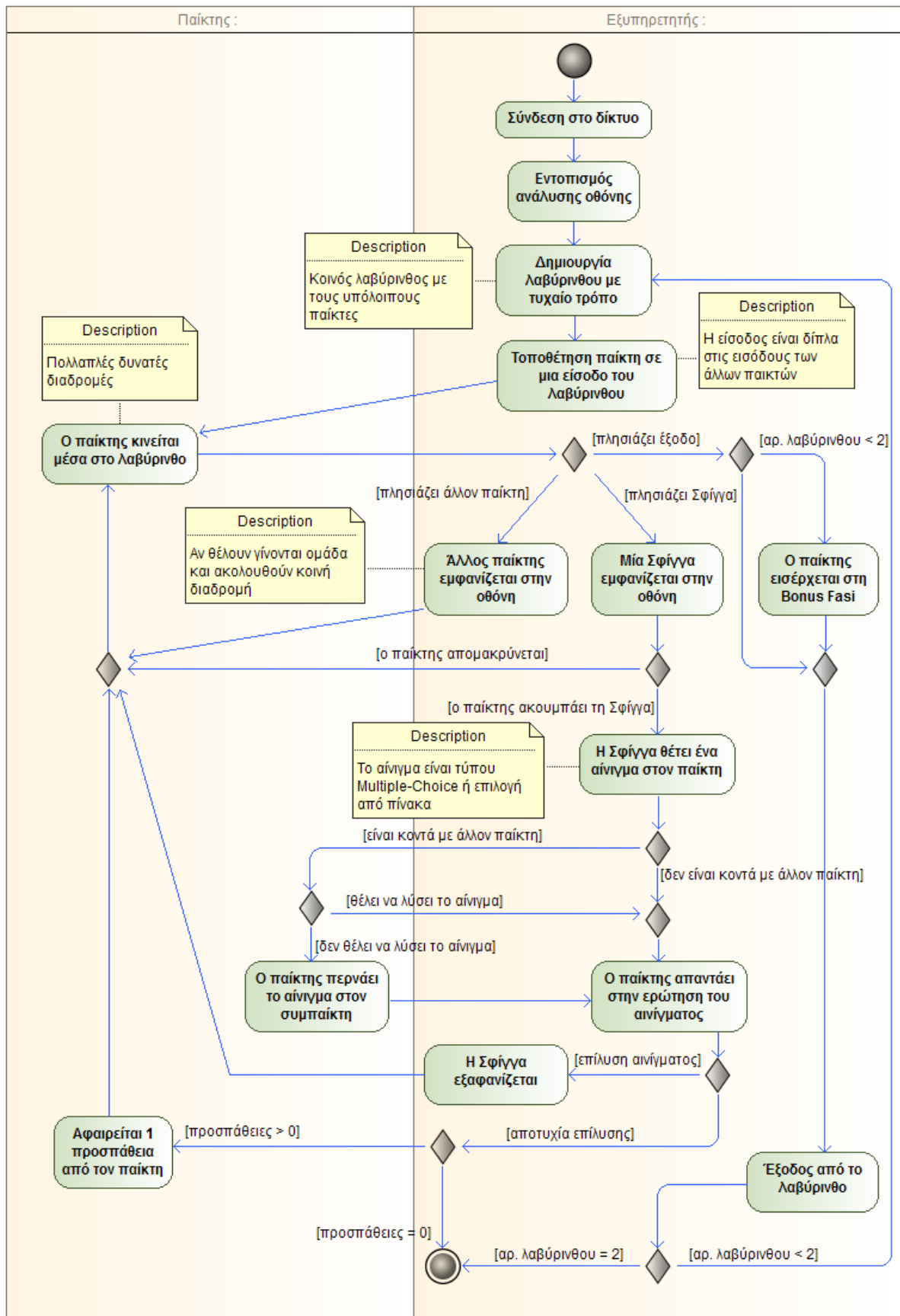


Σχήμα 2.2: UML Διάγραμμα Καταστάσεων.

Το κυκλικό σύμβολο στην αρχή αριστερά του σχεδιαγράμματος αναπαριστά την εκκίνηση της εφαρμογής και το σύμβολο στο τέλος δεξιά αναπαριστά την τελική κατάσταση/έξοδο από την εφαρμογή.

2.4.3 Διάγραμμα Δραστηριοτήτων (Activity Diagram)

Με το παρακάτω διάγραμμα περιγράφεται δυναμικά ο τρόπος έκθεσης των αλληλεπιδράσεων, εστιάζοντας στην εργασία. Είναι βασικά ένα διάγραμμα ροής που αναπαριστά τη ροή από τη μία δραστηριότητα στην άλλη και αντιπροσωπεύει τη συμπεριφορά του συστήματος από την άποψη του προγραμματιστή.



Σχήμα 2.3: UML Διάγραμμα Δραστηριοτήτων.

2.4.4 Διαγράμματα Κλάσεων (Class Diagrams)

Το διάγραμμα κλάσεων αντιπροσωπεύει τη στατική άποψη της εφαρμογής. Εδώ θα φανεί η δομή και η διάρθρωση του Λαβύρινθου. Οι πιθανές κλάσεις είναι τα ουσιαστικά από την περιγραφή των ζητούμενων του προβλήματος. Φυσικά, κάποια ουσιαστικά μπορούν να απορριφθούν, ανάλογα με το αν βρίσκονται στην περιοχή του προβλήματος ή όχι.

Τα ουσιαστικά είναι:

παιχνίδι	παίκτης	multiplayer
σύνδεση	bluetooth	wi-fi
έξυπνη κινητή συσκευή	ρόλος	εξυπηρετητής
πελάτης	μετακίνηση	λαβύρινθος
σύστημα	οθόνη αφής	touchscreen
οθόνη	ακολουθία	διεκπεραίωση
πλατφόρμα	δίκτυο	πρόγραμμα
ανάλυση	παράμετρος	στοιχείο
γραφικά	διαδρομή	τρόπος
είσοδος	έξοδος	τοποθεσία
Σφίγγα	αίνιγμα	εμφάνιση
ερώτηση	απάντηση	τύπος
πολλαπλή-επιλογή	multiple-choice	επίπεδο
δυσκολία	κατηγορία	Λογισμικό
Υλικό	Λειτουργικά Συστήματα	Δυναμικό Σύστημα
Ερώτηση Κρίσης	είδος	πίνακας
επιλογή	προσπάθεια	μήκος
πόντος	περίπτωση	μορφή
χρήστης	ερωταπάντηση	βάση δεδομένων
Bonus Φάση	κουτάκι	δικαίωμα
ρίξιμο	ζάρι	αριθμός

Κάποια από τα παραπάνω ουσιαστικά μπορεί τελικά να είναι ιδιότητες σε κλάσεις. Ιδιότητες θα είναι αυτά που περιγράφουν άλλα ουσιαστικά. Τέτοια ουσιαστικά είναι τα:

σύστημα (οθόνης αφής), ανάλυση (οθόνης), παράμετρος (στοιχείου απεικόνισης), επίπεδο (δυσκολίας), δυσκολία (λαβυρίνθου), κατηγορία (ερώτησης/ερωταπάντησης), επιλογή (απάντησης), δικαίωμα (κατηγορίας στο κουτάκι), μήκος (διαδρομής), αριθμός (ζαριού)

Επομένως, αυτές οι τελευταίες λέξεις απορρίπτονται από κλάσεις, αλλά θα αναλυθούν ξανά παρακάτω ως πιθανές ιδιότητες.

Το ουσιαστικό «ρίξιμο», το οποίο είναι από το ρήμα ρίχνω, αντί για κλάση θα αναλυθεί ξανά ως πιθανή συνάρτηση.

Επιπλέον, τα ουσιαστικά «Λογισμικό», «Υλικό», «Λειτουργικά Συστήματα», «Δυαδικό Σύστημα», «Ερώτηση Κρίσης» φαίνονται ως τιμές της ιδιότητας «κατηγορία», οπότε απορρίπτονται από κλάσεις.

Επίσης, απορρίπτονται μερικά ουσιαστικά που δεν περιγράφουν την κεντρική ιδέα του προβλήματος. Αυτά είναι τα «ακολουθία», «διεκπεραίωση», «τρόπος», «μορφή» και «περίπτωση».

Οι λέξεις «χρήστης», «συσκευή», «τύπος», «πολλαπλή-επιλογή» και «ερωταπάντηση» απορρίπτονται γιατί είναι συνώνυμες των «παίκτης», «πλατφόρμα», «είδος», «multiple-choice» και αίνιγμα αντίστοιχα.

Η «οθόνη» μπορεί να απορριφθεί, αφού μοιάζει περισσότερο ως μια μικρή λίστα από ιδιότητες, παρά για κλάση. Οι ιδιότητες που είναι πιθανό να προκύψουν μπορούν να ενσωματωθούν στην κλάση «πλατφόρμα».

Τα ουσιαστικά «προσπάθεια» και «πόντος» απορρίπτονται από κλάσεις, αφού περιγράφουν το ουσιαστικό «σκορ», το οποίο δεν υπάρχει στο κείμενο με τις απαιτήσεις, αλλά το προσθέτουμε σαν κλάση, γιατί αφορά γενικά τη σφαίρα του προβλήματος. Παρόλα αυτά, οι απορριφθείσες λέξεις «προσπάθεια» και «πόντος» είναι πιθανές ιδιότητες του «σκορ», γιατί το περιγράφουν.

Το «δίκτυο» μοιάζει περισσότερο με πακέτο java package και θα εξεταστεί ως τέτοιο.

Το «multiplayer» ταιριάζει πιο πολύ ως ιδιότητα και θα εξεταστεί ως τέτοια.

Τέλος το ουσιαστικό «πρόγραμμα» περιγράφει ολόκληρη την εφαρμογή, κι έτσι δε θα υλοποιηθεί σαν απλή κλάση.

Τα ουσιαστικά – και μέλλουσες κλάσεις – που απομένουν είναι τα εξής:

παιχνίδι	παίκτης	σύνδεση	bluetooth
wi-fi	ρόλος	εξυπηρετητής	πελάτης
μετακίνηση	λαβύρινθος	touchscreen	οθόνη
πλατφόρμα	στοιχείο	γραφικά	διαδρομή
είσοδος	έξοδος	τοποθεσία	Σφίγγα
αίνιγμα	εμφάνιση	ερώτηση	απάντηση
είδος	multiple-choice	πίνακας	βάση δεδομένων
σκορ	Bonus Φάση	κουτάκι	ζάρι

Κατά την υλοποίηση της εφαρμογής, απαιτήθηκε να γίνουν μερικές τροποποιήσεις, στις οποίες οδήγησαν οι ιδιαιτερότητες της αρχιτεκτονικής του Android, αλλά και η λογική του μηχανισμού υλοποίησης ενός παιχνιδιού. Έτσι, αυτό είχε ως αποτέλεσμα:

- Η κλάση «τοποθεσία» διαπιστώνεται ότι αντιστοιχεί στις υπάρχουσες κλάσεις του Android SDK «Point» για ακεραίους (integers) και «PointF» για δεκαδικούς απλής ακρίβειας (floats), με τις οποίες τελικά αντικαθίσταται.
- Στη Java σε κάθε εφαρμογή η εκτέλεση ξεκινάει με την μέθοδο «main()». Αντίθετα, στη Java για Android η εφαρμογή ξεκινάει μέσω της μεθόδου «onCreate()» μιας κύριας Activity class. Αντίστοιχα για τις ανάγκες της εφαρμογής «Λαβύρινθος» προστίθεται η κλάση «Main», όπου θα υλοποιείται η «onCreate()».
- Η υλοποίηση του δικτύου μπορεί να γίνει με δύο τρόπους. Κατά τον πρώτο τρόπο, προκαθορίζεται η μία συσκευή να λειτουργήσει ως server και άλλη ως client. Κατά το δεύτερο τρόπο, ξεκινάνε και οι δύο συσκευές ως servers και στη συνέχεια, η συσκευή που θα επιλέξει πρώτη να συνδεθεί στην άλλη, θα μετατραπεί σε client. Επιλέχθηκε τελικά ο δεύτερος τρόπος και οι υλοποιήσεις server και client ενσωματώθηκαν στην κλάση «Bluetooth». Συνεπώς, οι κλάσεις «εξυπηρετητής» και «πελάτης» παραλείπονται, όπως παραλείπεται και η κλάση-γονέας τους, «ρόλος», γιατί ο ρόλος «είναι» εξυπηρετητής ή πελάτης (σχέση «είναι»).
- Δημιουργείται μία κλάση «Assets», για να διαχειριστεί την αποκωδικοποίηση και εισαγωγή στοιχείων από εξωτερικές πηγές, δηλαδή τα αρχεία εικόνας, ήχου και γραμματοσειράς. Το παραπάνω όνομα επιλέγεται, επειδή τα παραπάνω αρχεία απαιτείται από το Android SDK να συγκεντρωθούν στον προκαθορισμένο φάκελο Assets.

- Δημιουργείται εξειδικευμένη κλάση «Accelerometer» για να αξιοποιήσει τον αισθητήρα επιτάχυνσης που διαθέτουν οι περισσότερες κινητές συσκευές.
- Η μετακίνηση του παίκτη στο λαβύρινθο γίνεται μέσω αντικειμένων των κλάσεων «TouchScreen» και «Accelerometer», επομένως το ουσιαστικό «μετακίνηση» δεν είναι απαραίτητο να γίνει κλάση και απορρίπτεται. Ωστόσο η φράση μπορεί να διατυπωθεί ως «ο παίκτης μετακινείται στο λαβύρινθο» κι έτσι το ρήμα «μετακινείται» να εξεταστεί μεταγενέστερα σαν πιθανή συνάρτηση.
- Για τις ανάγκες υλοποίησης ενός παιχνιδιού γενικότερα, απαιτούνται μέθοδοι που να εντοπίζουν συγκρούσεις (collisions) μεταξύ των στοιχείων με τον ταχύτερο δυνατό τρόπο. Αυτόν το σκοπό μπορεί να εξυπηρετήσει μια κλάση που θα τοποθετεί τα διασπαρμένα στοιχεία σε ένα πλέγμα διαστάσεων ίσων με το χώρο του παιχνιδιού, όπου τα στοιχεία θα κατακερματίζονται στις περιοχές του πλέγματος. Το όνομα της κλάσης ορίζεται ως «SpatialHashGrid», το οποίο αποτελεί έναν περιγραφικό τίτλο του αλγορίθμου στην αγγλική γλώσσα.
- Κατά την υλοποίηση του παιχνιδιού, γραφικά στοιχεία όπως τα κουμπιά (buttons) που παρέχει το Android SDK δεν είναι διαθέσιμα, καθώς χρησιμοποιείται μια κενή προβολή SurfaceView για τη σχεδίαση των γραφικών. Γι' αυτόν το λόγο προστίθεται μία κλάση «UIButton», προκειμένου να υλοποιηθούν προσαρμοσμένα γραφικά κουμπιά.
- Όπως είδαμε παραπάνω στο διάγραμμα καταστάσεων, δημιουργήθηκε η ανάγκη να μπορεί ο κάθε παίκτης να επιλέξει ένα εικονίδιο που θα τον αντιπροσωπεύει. Με αυτόν τον τρόπο, οι παίκτες θα μπορούν ξεχωρίζουν το χαρακτήρα τους στο δικτυακό παιχνίδι, ενώ ταυτόχρονα είναι πιθανό να προσθέτει περισσότερη ικανοποίηση και ψυχαγωγία στο χρήστη. Αυτή η κατάσταση του παιχνιδιού, όπου θα γίνεται η επιλογή της εικόνας του παίκτη (avatar), μπορεί να υλοποιηθεί με μια κλάση «Avatar».
- Δημιουργείται αντίστοιχα μια κλάση «GameOver», προκειμένου να αναπαρασταθεί η τελική οθόνη του παιχνιδιού.

Με βάση τις τελευταίες παρατηρήσεις και με μερικές μετονομασίες στα παραπάνω ουσιαστικά (π.χ. παίκτης=Player), προκύπτουν οι εξής κλάσεις:



Σχήμα 2.4: Κλάσεις.

Packages

Οι κλάσεις μπορούν να χωριστούν σε ομάδες με βάση την απλή λογική, τα αποκαλούμενα packages στη Java.

Το πλαίσιο εργασίας (framework) του παιχνιδιού αποτελείται από τα Activities εκκίνησης και εκτέλεσης, καθώς και τις κλάσεις που αφορούν τον κύριο μηχανισμό του. Η εκκίνηση γίνεται με τη «Main», η αρχικοποίηση με την «Platform», ενώ η κύρια μηχανή είναι η «Game», σε συνδυασμό με τις



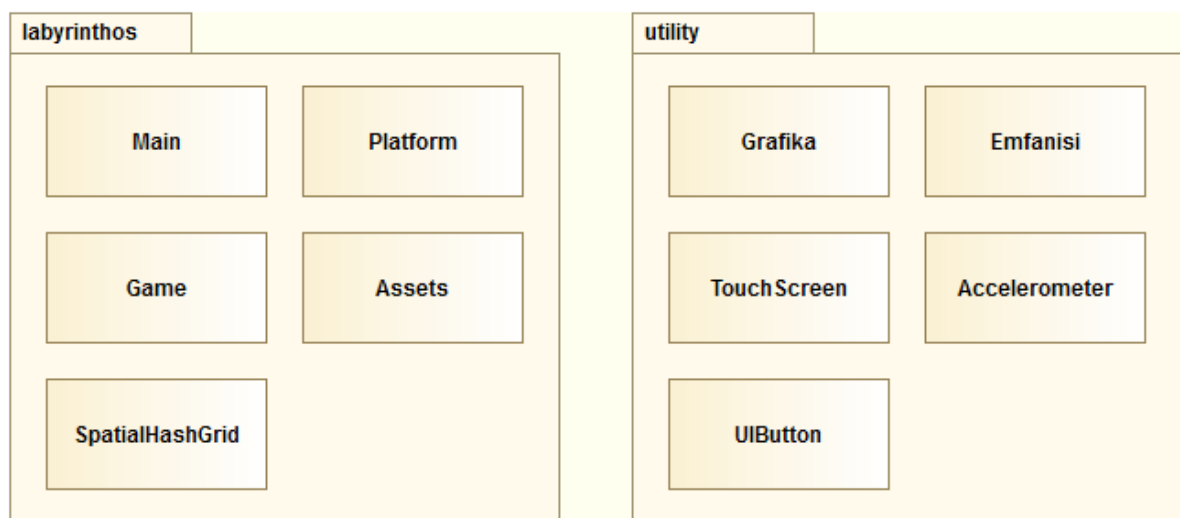
Σχήμα 2.5: Package labyrinthos

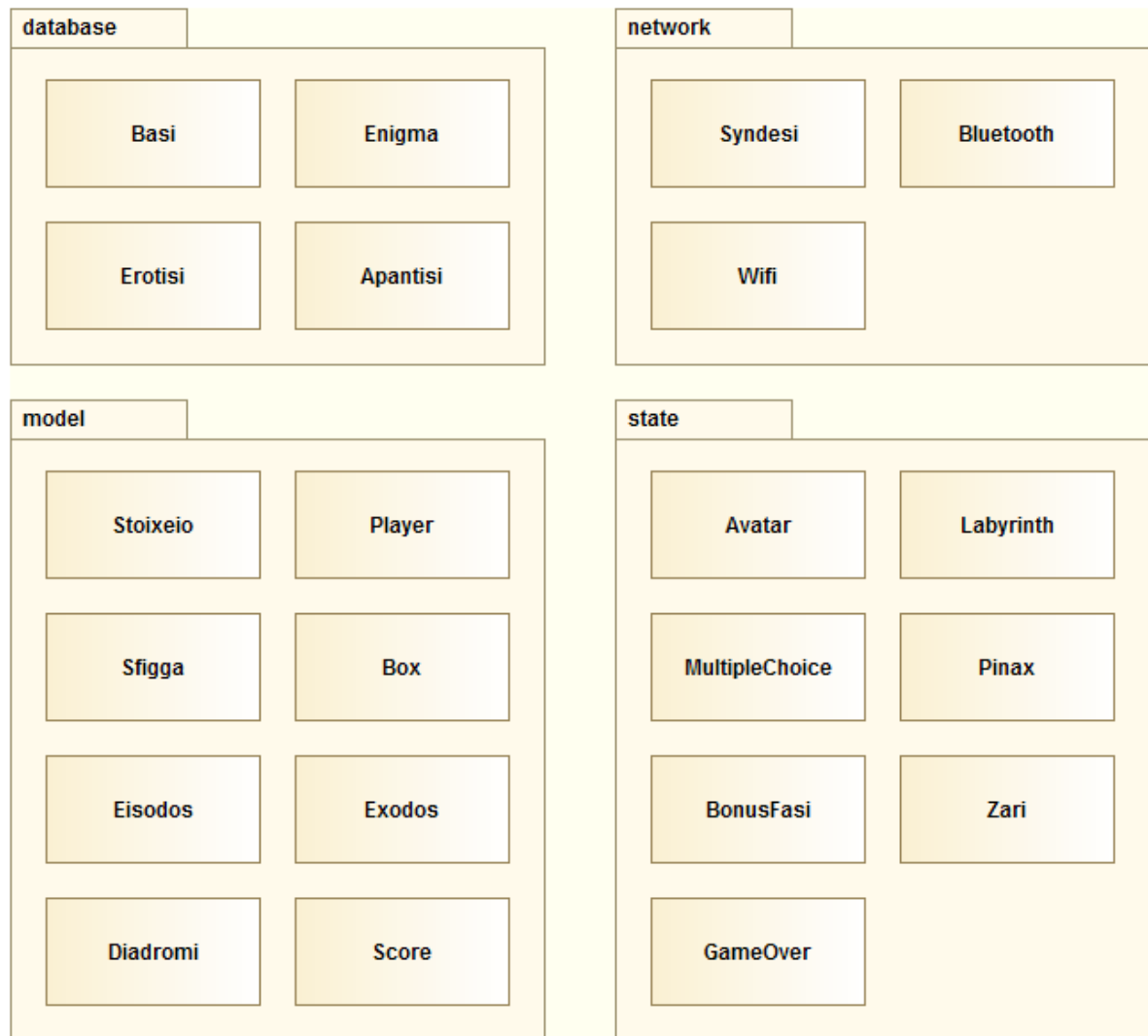
κλάσεις «Assets» και «SpatialHashGrid». Αυτά περιγράφουν το παιχνίδι «Λαβύρινθος», επομένως οι παραπάνω κλάσεις αποτελούν το package «labyrinthos».

Από τις υπόλοιπες κλάσεις:

- Οι «Grafika», «Emfanisi», «TouchScreen», «Accelerometer» και «UIButton» περιέχουν βοηθήματα από το Android SDK για την εμφάνιση και τη διασύνδεση με τη κινητή πλατφόρμα, οπότε ομαδοποιούνται στο package «utility».
- Οι κλάσεις «Stoixeiio», «Player», «Sfigga», «Box», «Eisodos», «Exodos», «Diadromi», «Score», αποτελούν τα μοντέλα των διαφόρων στοιχείων του Λαβύρινθου και έχει νόημα να σχηματίσουν το package «model».
- Οι κλάσεις «Avatar», «Labyrinth», «MultipleChoice», «Pinax», «BonusFasi», «Zari» και «GameOver», είναι οι οθόνες των καταστάσεων οι οποίες απεικονίζονται διαδοχικά στο παιχνίδι, κατά την εκτέλεση. Μπαίνουν στο package «state».
- Οι κλάσεις «Basi», «Enigma», «Erotisi», «Arantsi» αφορούν τα αποθηκευμένα δεδομένα ερωτοαπαντήσεων που υποβάλλονται στο χρήστη και δημιουργούν την ομάδα package «database».
- Τέλος, μία ακόμη ομάδα φέρεται να είναι το package «network», όπως έχει προαναφερθεί για το ουσιαστικό «δίκτυο», με το οποίο σχετικά είναι τα «Syndesi», «Bluetooth», «Wifi».

Όλα packages παρατίθενται συγκεντρωτικά στο παρακάτω διάγραμμα:





Σχήμα 2.6: Διάγραμμα Κλάσεων - Packages.

Από τα παραπάνω λείπει η κλάση «Eidos», που παραμένει μόνη της προς το παρόν.

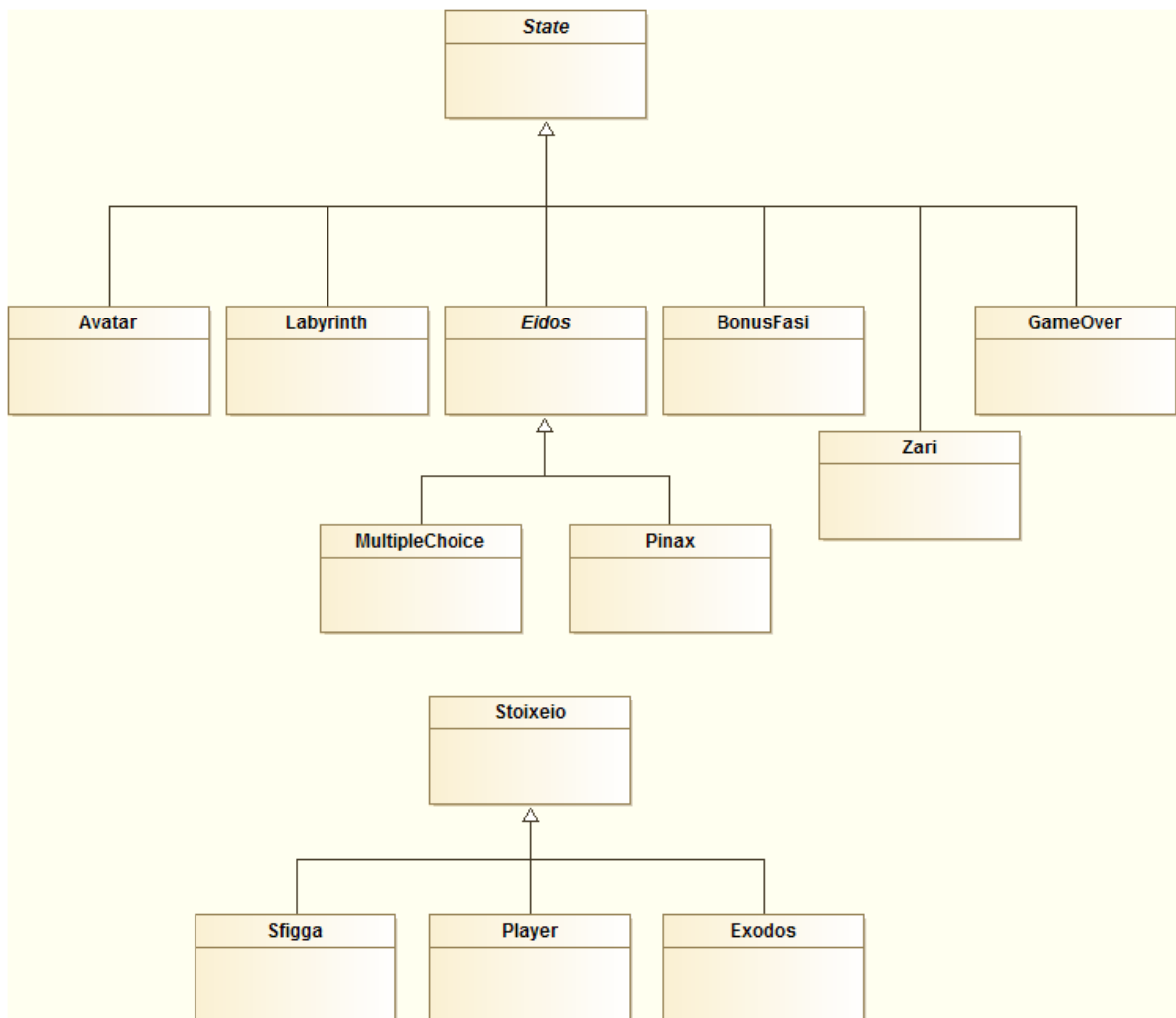
Κληρονομικότητα

Η σχέση του «είναι (ένα είδος)» μεταξύ των κλάσεων περιγράφει την κληρονομικότητα. Οι σχέσεις αυτές είναι:

- Από το package «model», οι κλάσεις «Player», «Sfigga», «Exodos» και «Stoixeio» είναι στοιχεία με πιθανά κοινά χαρακτηριστικά, αλλά και οπτική αναπαράσταση στο παιχνίδι με γραφικά. Επομένως η κλάση «Stoixeio» μπορεί να γενικευτεί και να θεωρηθεί ως γονέας και οι «Player», «Sfigga», «Exodos» ως παιδιά της.

- Από την περιγραφή του κειμένου των απαιτήσεων, τα «MultipleChoice» και «Pinax» είναι «Eidos». Επομένως, τα «MultipleChoice», «Pinax» είναι παιδιά του «Eidos». Επιπλέον, η κλάση «Eidos» είναι αφηρημένη (abstract), γιατί από την περιγραφή φαίνεται ότι δεν συγκεκριμενοποιείται, καθώς είναι άλλοτε «MultipleChoice» και άλλοτε «Pinax». Έτσι, υπάρχει νόημα να συμπεριληφθεί στο package «state», όπου ανήκουν και «MultipleChoice» και «Pinax».
- Στο package «state» όλες οι κλάσεις είναι καταστάσεις (states). Κάθε κατάσταση είναι μια διαφορετική οθόνη στο παιχνίδι. Ορίζεται λοιπόν μια νέα αφηρημένη κλάση «State» ως γονέας των «Avatar», «Labyrinth», «Eidos» («MultipleChoice» ή «Pinax»), «BonusFasi», «Zari», «GameOver».

Το παρακάτω διάγραμμα κλάσεων απεικονίζει τις σχέσεις κληρονομικότητας στις κλάσεις της εφαρμογής:



Σχήμα 2.7: Διάγραμμα Κλάσεων - Κληρονομικότητα.

Ιδιότητες

Για να εντοπίσουμε τις ιδιότητες των κλάσεων, βρίσκουμε κατ' αρχήν τις λέξεις – ουσιαστικά και επίθετα – και τις φράσεις στο κείμενο με τα ζητούμενα, που περιγράφουν τις κλάσεις. Τα ουσιαστικά που είναι πιθανές ιδιότητες είχαν εντοπιστεί προηγούμενα και είναι:

Ουσιαστικά

σύστημα	ανάλυση
παράμετρος	επίπεδο
κατηγορία	επιλογή
δικαίωμα	μήκος
προσπάθεια	πόντοι
δυσκολία	αριθμός
multiplayer	

Τα επίθετα που είναι πιθανές ιδιότητες είναι:

Επίθετα

ένας ή περισσότερος	ασύρματος
υποστηριζόμενος	τυχαίος
κοινός	διαφορετικός
εναλλακτικός	σίγουρος
σωστός	διάφορος
επιτυχής	επόμενος
παραγόμενος	αυξημένος

Από τα παραπάνω επίθετα δεχόμαστε μόνο τα

πλήθος (=ένας ή περισσότερος)

τυχαία (ερώτηση)

τυχαίο (ζάρι)

σίγουρος

σωστό

επιτυχής

επόμενος

καθώς είναι τα μόνα, τα οποία βρίσκονται πραγματικά στο χώρο του προβλήματος και περιγράφουν κάποιες από τις παραπάνω κλάσεις - το επίθετο «ασύρματος» για παράδειγμα αφορά ιδιότητα του λειτουργικού συστήματός μας κι όχι το πρόβλημά μας και γι' αυτόν το λόγο εξαιρείται.

Στα παραπάνω επίθετα - ιδιότητες προστίθενται και τα παραπάνω ουσιαστικά λέξεις-ιδιότητες, οι οποίες περιγράφουν γενικά την κάθε κλάση. Κατά εξαίρεση, η ιδιότητα «ανάλυση» είναι πρακτικότερο να σπάσει στις δύο υπό-ιδιότητες «μήκος» και «ύψος» και επειδή θα χαρακτηρίζουν τη γηγενή ανάλυση του παιχνιδιού τις ονοματίζουμε στα αγγλικά ως «GAME_WIDTH» και «GAME_HEIGHT».

Συναρτήσεις - Συσχετίσεις

Τα ρήματα από το κείμενο με τα ζητούμενα είναι οι πιθανές συναρτήσεις (λειτουργίες) των κλάσεων. Αυτά είναι:

υποστηρίζει	ακολουθήσουν (κοινή διαδρομή)
επιτυγχάνεται	περνάει (την ερώτηση στον άλλο)
επιτευχθεί	μπορούν (ερωτήσεις αινίγματος)
παίζουν (ρόλο)	έχει
συνδέονται	απαντήσει (σωστά)
γίνεται	αθροίζονται
απεικονίζεται	ανακαλυφθούν
εντοπίζει	καταγράφονται (πόντοι)
ρυθμίζει	αναγράφονται (στην οθόνη)
χωρέσουν	βρουν
δημιουργείται (λαβύρινθος)	εισέρχεται
δημιουργούνται (διαδρομές)	συνεχίζουν
είναι (τύπος αινίγματος/είδος)	τελειώνει
υπάρχει	Δίνεται (αίνιγμα)
τοποθετείται (ο παίκτης)	καλείται να βρει
δημιουργούνται (είσοδοι/έξοδοι)	κερδίζει (προσπάθεια)

εμφανίζονται (Σφίγγες)	προβάλλονται (απαντήσεις)
κάνουν (ερώτηση οι Σφίγγες)	καλείται να απαντήσει
επιλέγει	προβάλλονται (αινίγματα)
πέφτει	περιλαμβάνει
λύσει (αίνιγμα)	περιέχει
περάσει (ο χρήστης τη Σφίγγα)	χάνει (προσπάθεια)
συναντηθούν (παίκτες)	μετακινείται

Στα παραπάνω ρήματα προσθέτουμε το ρήμα «ρίχνει» από το ουσιαστικό «ρίξιμο», το οποίο προηγουμένως δεν επιλέχθηκε σαν κλάση, καθώς παραπέμπει περισσότερο σε λειτουργία.

Μπορούμε να μειώσουμε το παραπάνω πλήθος ρημάτων-λειτουργιών, αφαιρώντας τις συνώνυμες λέξεις: επιτευχθεί = *επιτυγχάνεται*, καλείται να απαντήσει = *απαντάει*

Το ρήμα «υποστηρίξει» απορρίπτεται από συνάρτηση, αφού αφορά το ουσιαστικό «παιχνίδι», που έχει απορριφθεί από κλάση πρωτύτερα.

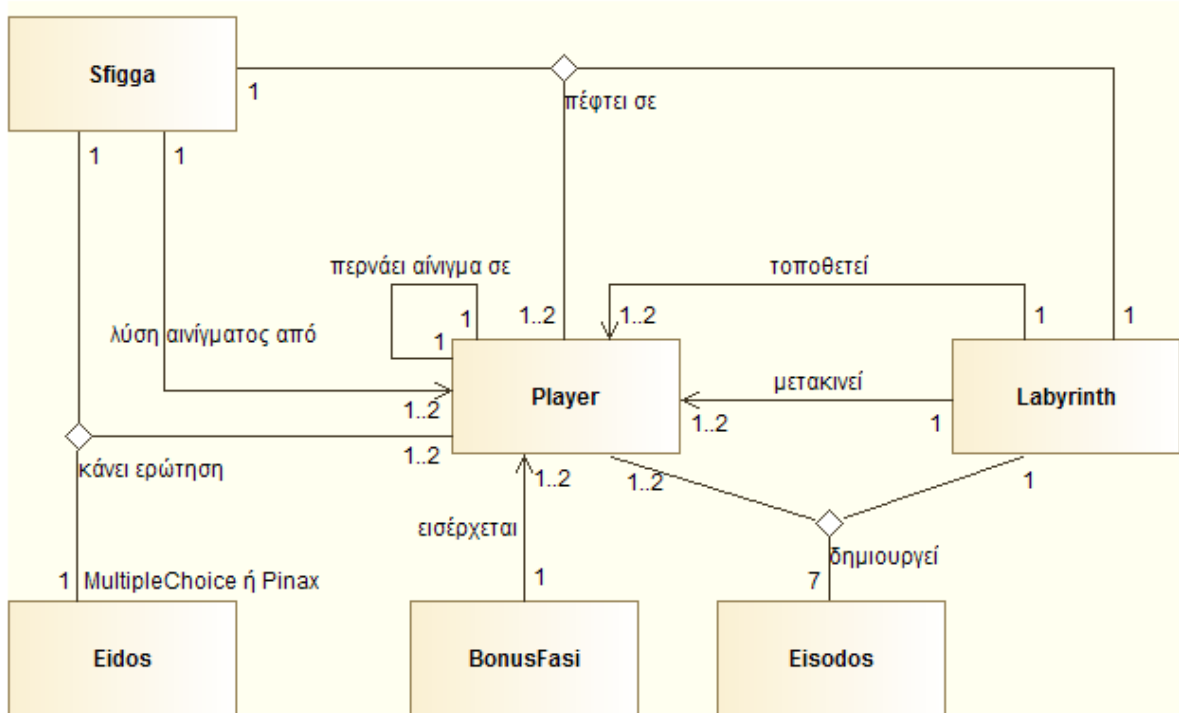
Η φράση με τα ρήματα «καταφέρουν να συναντηθούν» μπορεί να αντικατασταθεί με το ρήμα «συναντάει».

Το ρήμα «είναι» παραπέμπει σε κληρονομικότητα κι όχι σε μέθοδο. Το Είδος «είναι» MultipleChoice ή Pinax, όπως έχει αναλυθεί και παραπάνω.

Ακολούθως, θα δημιουργηθούν οι συσχετίσεις μεταξύ των κλάσεων, εστιάζοντας σε μερικές από τις σημαντικότερες κλάσεις του προβλήματος. Οι συσχετίσεις της κλάσης Player με τις άλλες κλάσεις μπορούν να εντοπιστούν εξετάζοντας τα ρήματα και τις φράσεις που μπορούν να τις συσχετίσουν μεταξύ τους, δηλαδή:

Labyrinth	δημιουργεί	Player	σε 1 από 7	Eisodos
Player	μετακινείται στο	Labyrinth		
Player	τοποθετείται στο	Labyrinth		

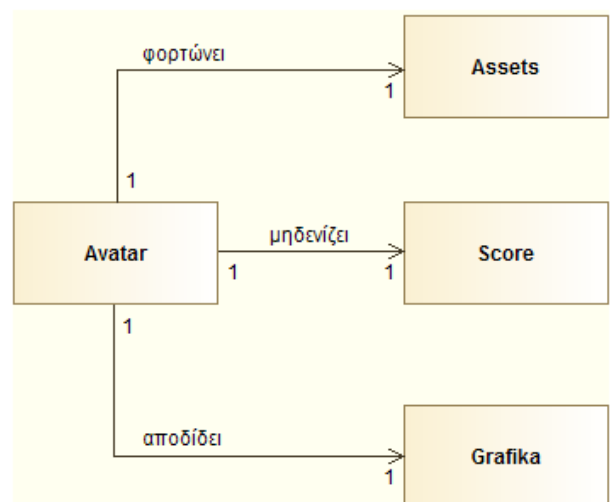
Player	πέφτει σε	Sfigga	του	Labyrinth
Sfigga	κάνει ερώτηση στον	Player	τύπου	Eidos
Sfigga	λύση αινίγματος από	Player		
Player	περνάει αίνιγμα σε	Player (άλλον)		
Player	εισέρχεται σε	BonusFasi		



Σχήμα 2.8: Διάγραμμα Κλάσεων - Συσχετίσεις Player

Οι συσχετίσεις της κλάσης Avatar:

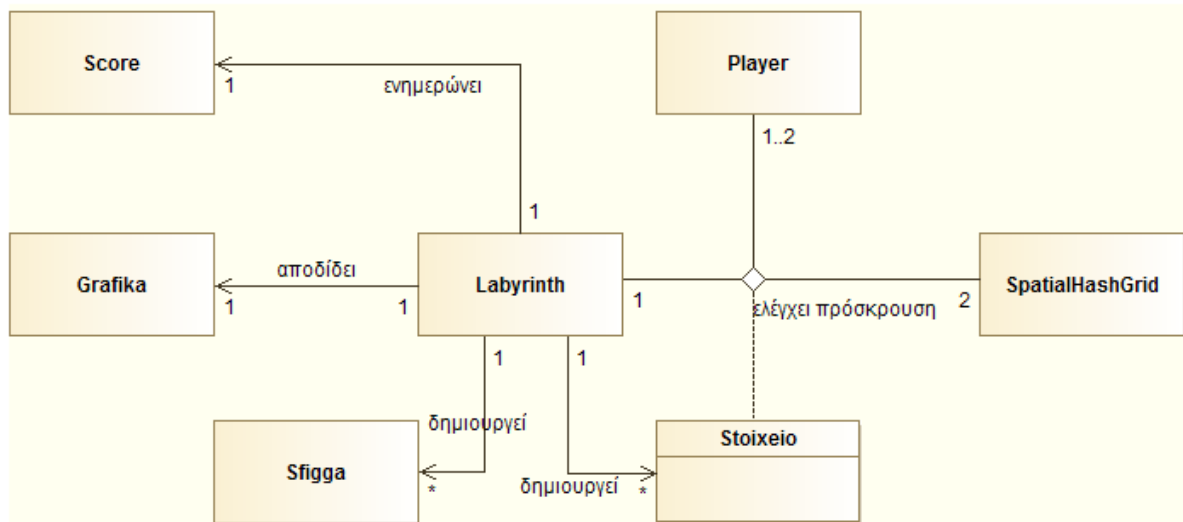
Avatar	φορτώνει	Assets
Avatar	μηδενίζει	Score
Avatar	αποδίδει (render)	Grafika



Σχήμα 2.9: Διάγραμμα Κλάσεων - Συσχετίσεις Avatar

Οι συσχετίσεις της κλάσης Labyrinth:

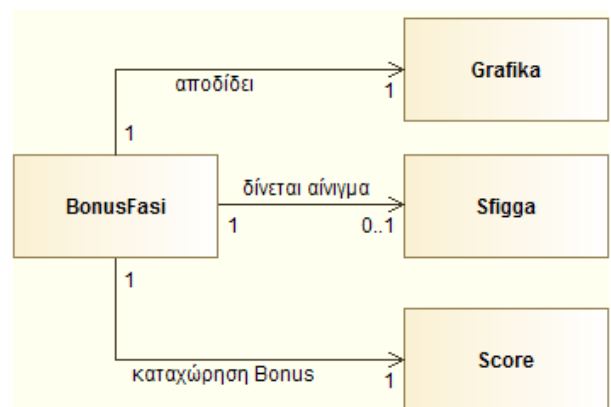
Labyrinth	ενημερώνει	Score
Labyrinth	αποδίδει (render)	Grafika
Labyrinth	δημιουργεί	Stoixeio (πολλά)
Labyrinth	δημιουργεί	Sfigga (πολλές)
Labyrinth	ελέγχει πρόσκρουση	Player με Stoixeio στο SpatialHashGrid



Σχήμα 2.10: Διάγραμμα Κλάσεων - Συσχετίσεις Labyrinth

Οι συσχετίσεις της κλάσης BonusFasi:

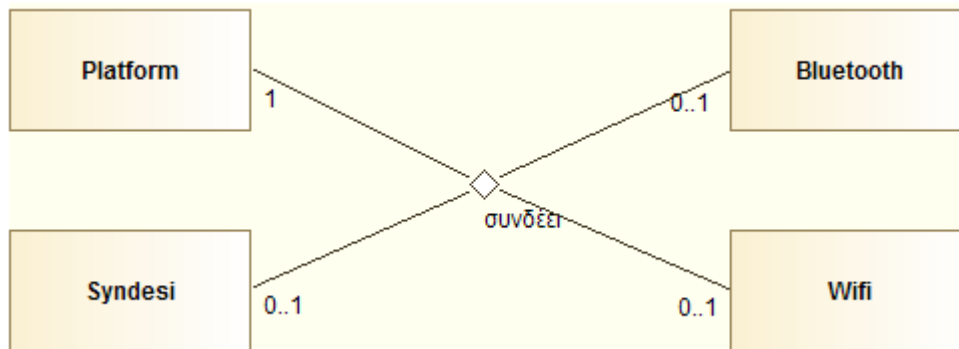
BonusFasi	αποδίδει (render)	Grafika
BonusFasi	δίνεται αίνιγμα	Sfigga
BonusFasi	καταχώρηση Bonus	Score



Σχήμα 2.11: Διάγραμμα Κλάσεων - Συσχετίσεις BonusFasi

Οι συσχετίσεις της κλάσης Platform:

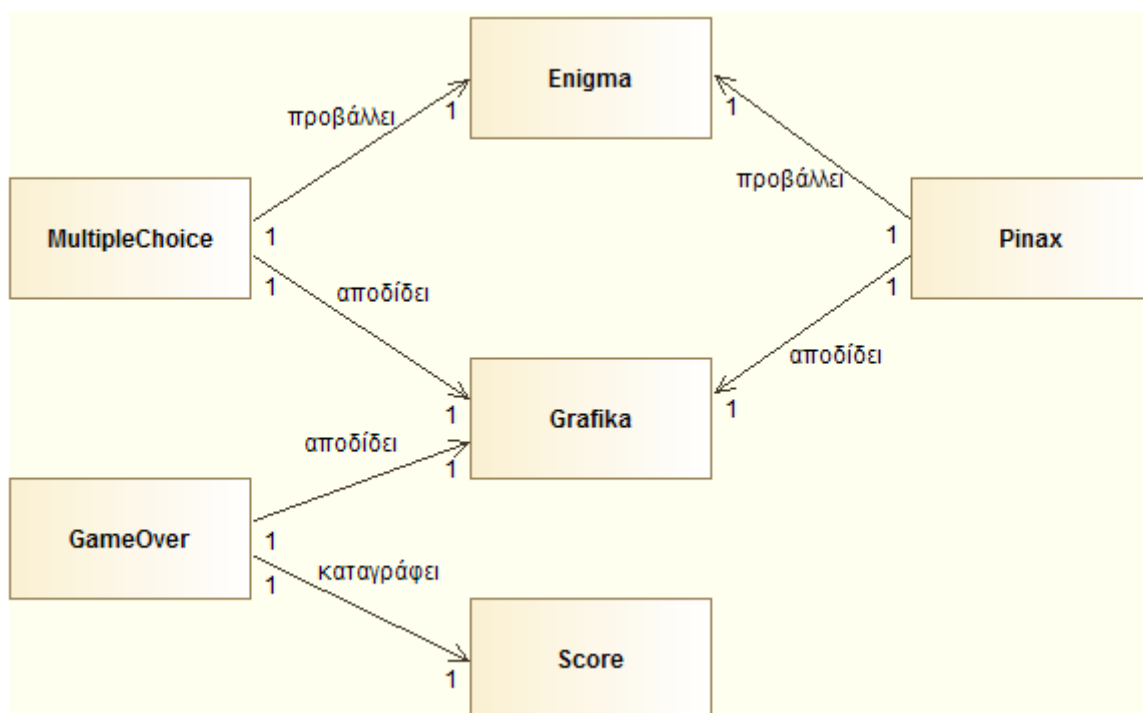
Platform	καλεί	Syndesi	για να συνδεθεί με	Bluetooth ή Wifi
----------	-------	---------	--------------------	------------------



Σχήμα 2.12: Διάγραμμα Κλάσεων - Συσχετίσεις Platform

Οι συσχετίσεις των υπόλοιπων κλάσεων:

MultipleChoice	προβάλλει	Enigma
MultipleChoice	αποδίδει (render)	Grafika
Pinax	προβάλλει	Enigma
Pinax	αποδίδει (render)	Grafika
GameOver	καταγράφει	Score
GameOver	αποδίδει (render)	Grafika

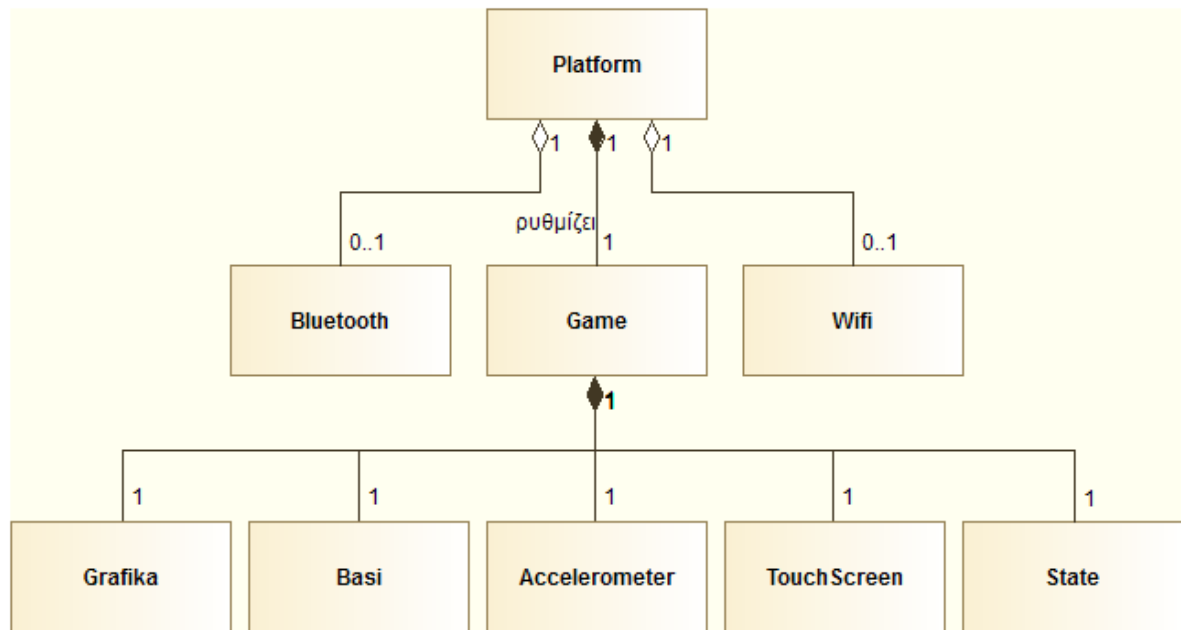


Σχήμα 2.13: Διάγραμμα Κλάσεων - Λοιπές Συσχετίσεις

Συσσωματώσεις και Συνθέσεις (Aggregations and Compositions)

Το επόμενο βήμα είναι να βρεθούν κλάσεις οι οποίες είναι συστατικά άλλων κλάσεων. Αυτό μπορεί να πραγματοποιηθεί συνδυάζοντας ρήματα όπως τα «έχει», «αποτελείται από» και «περιλαμβάνει» με τα ονομάτων των κλάσεων, για τη συγκρότηση λογικών προτάσεων. Έτσι, οι συσσωματώσεις και συνθέσεις που προκύπτουν είναι:

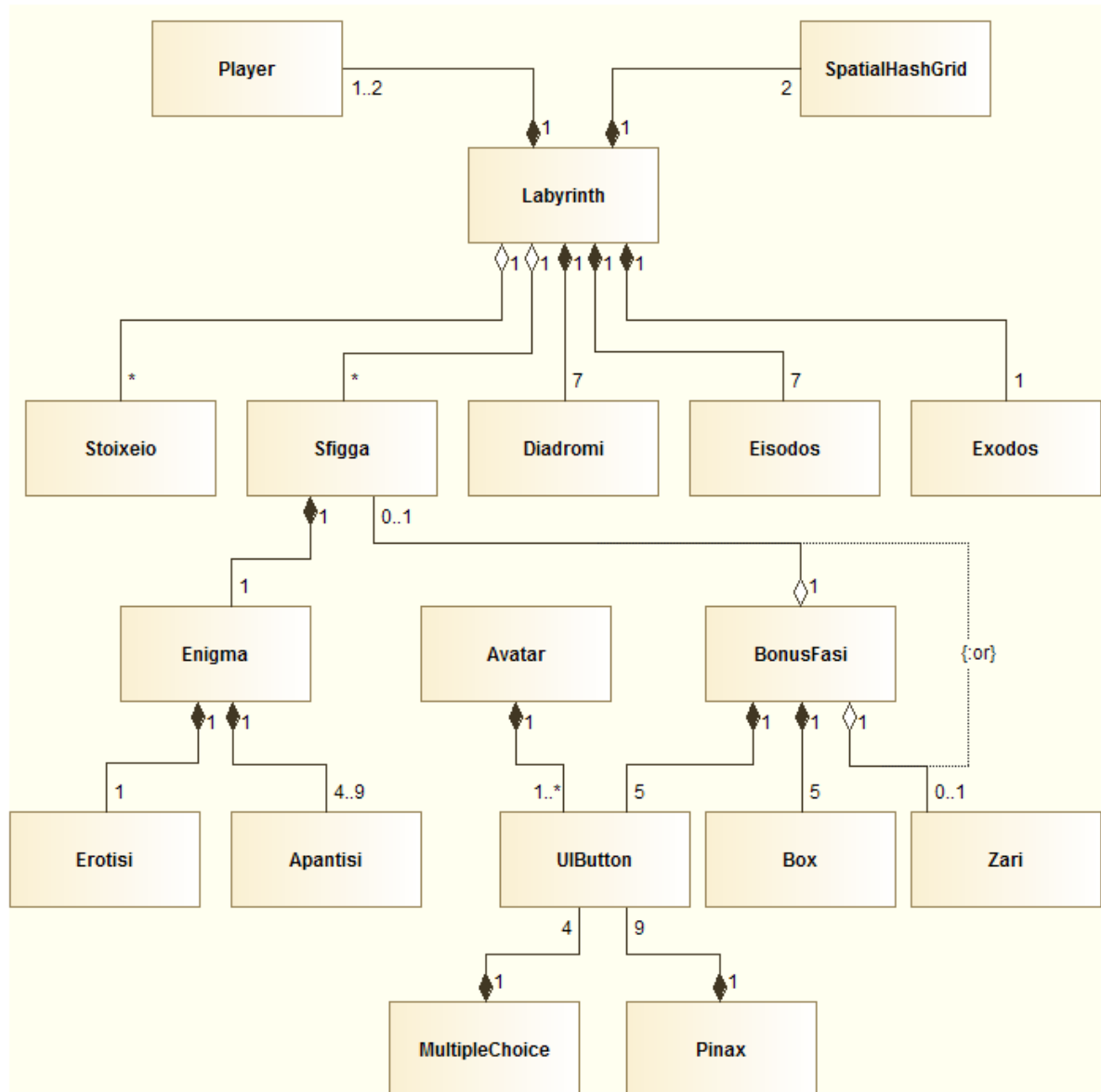
- Η πλατφόρμα («Platform») έχει και ρυθμίζει ένα παιχνίδι («Game»).
- Η πλατφόρμα («Platform») μπορεί να περιλαμβάνει και σύνδεση «Bluetooth» ή «Wifi».
- Το παιχνίδι («Game») αποτελείται από τα γραφικά («Grafika»), την εξωτερική βάση δεδομένων με τα αινίγματα («Basi»), τους αισθητήρες εισόδου («Accelerometer» και «TouchScreen»), καθώς και την τρέχουσα οθόνη κατάστασης («State») του παιχνιδιού.



Σχήμα 2.14: Διάγραμμα Κλάσεων - Σύνθεση Platform

- Ο λαβύρινθος («Labyrinth») μπορεί να περιλαμβάνει σφίγγες («Sfigga») και άλλα στοιχεία («Stoicheio»), όπως προσπάθειες/ζωές.
- Ο λαβύρινθος («Labyrinth») αποτελείται από τους παίκτες («Player»), τα πλέγματα κατακερματισμού των διαφόρων ειδών στοιχείων («SpatialHashGrid»), τις διαδρομές («Diadromi»), τις εισόδους («Eisodos») και την μία έξοδο («Exodos»).
- Η σφίγγα («Sfigga») έχει ένα αίνιγμα («Enigma»).
- Το αίνιγμα («Enigma») αποτελείται από μία ερώτηση («Erotisi») και τέσσερις ή εννιά απαντήσεις («Apantisi»).
- Η μπόνους φάση («BonusFasi») μπορεί να περιλαμβάνει μία σφίγγα («Sfigga») ή ένα ζάρι («Zari»).
- Η μπόνους φάση («BonusFasi») έχει πέντε κουτιά («Box») και πέντε γραφικά κουμπιά («UIButton»).
- Η οθόνη επιλογής εικόνας παίκτη («Avatar») αποτελείται από πολλά γραφικά κουμπιά με τις διαθέσιμες εικόνες («UIButton»).

- Η οθόνη πολλαπλών επιλογών («MultipleChoice») περιέχει τέσσερα γραφικά κουμπιά - απαντήσεις («UIButton»).
- Η οθόνη επιλογών από πίνακα («Pinax») περιέχει εννιά γραφικά κουμπιά - απαντήσεις («UIButton»).

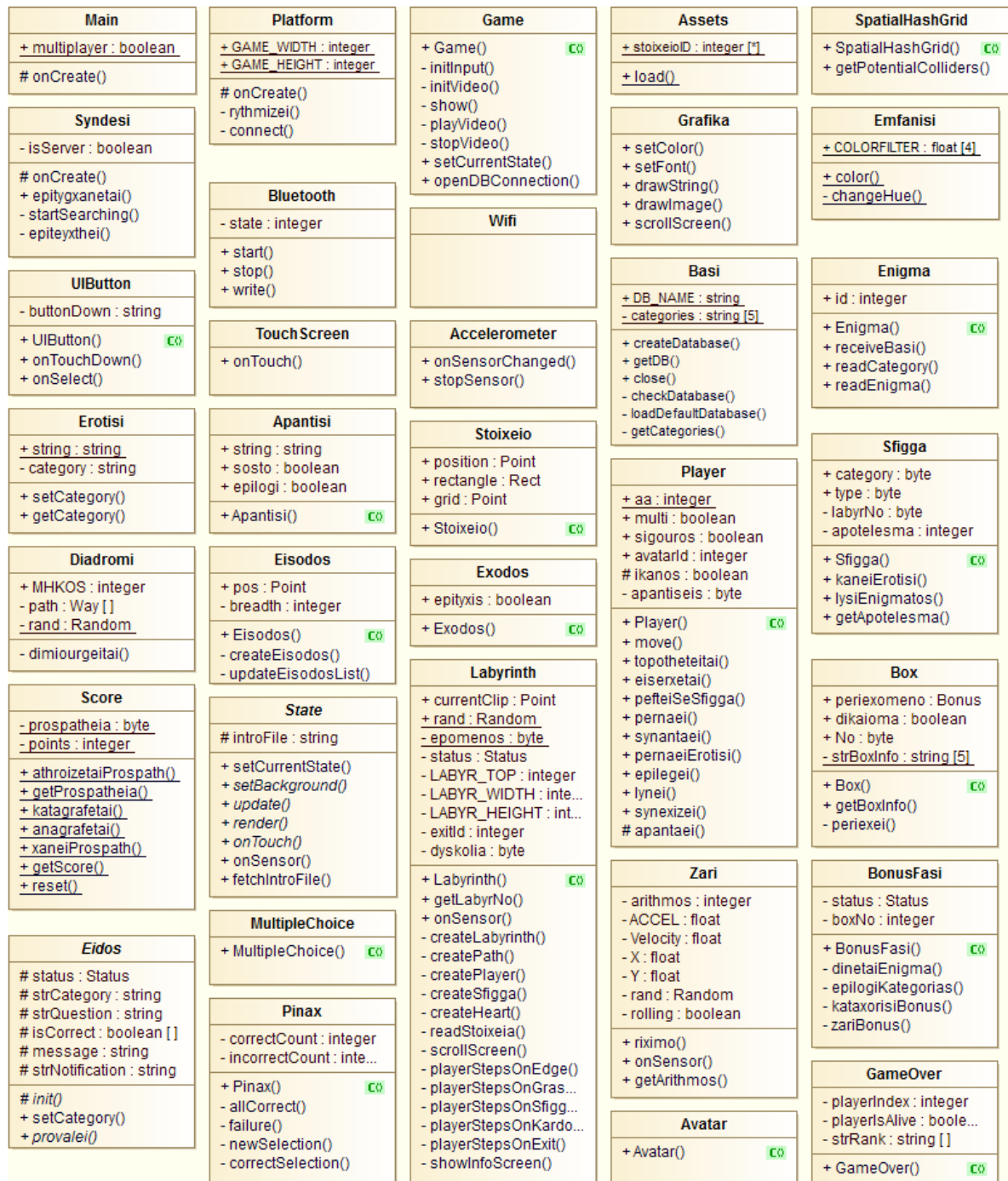


Σχήμα 2.15: Διάγραμμα Κλάσεων - Σύνθεσεις και Συσσωματώσεις

Συγκεντρωτικό διάγραμμα κλάσεων

Στο παρακάτω διάγραμμα κλάσεων, φαίνονται συγκεντρωτικά όλες οι κλάσεις της εφαρμογής μαζί με τις ιδιότητές τους και τις συναρτήσεις τους, όπως έχουν διαμορφωθεί έως τώρα. Έχουν μετονομασθεί κάποιες λέξεις με συνώνυμés τους ή αντικατασταθεί με αγγλικές, ενώ έχουν

τοποθετηθεί και κάποιες επιπλέον ιδιότητες, που δίνουν την περιγραφή των αντίστοιχων κλάσών τους.

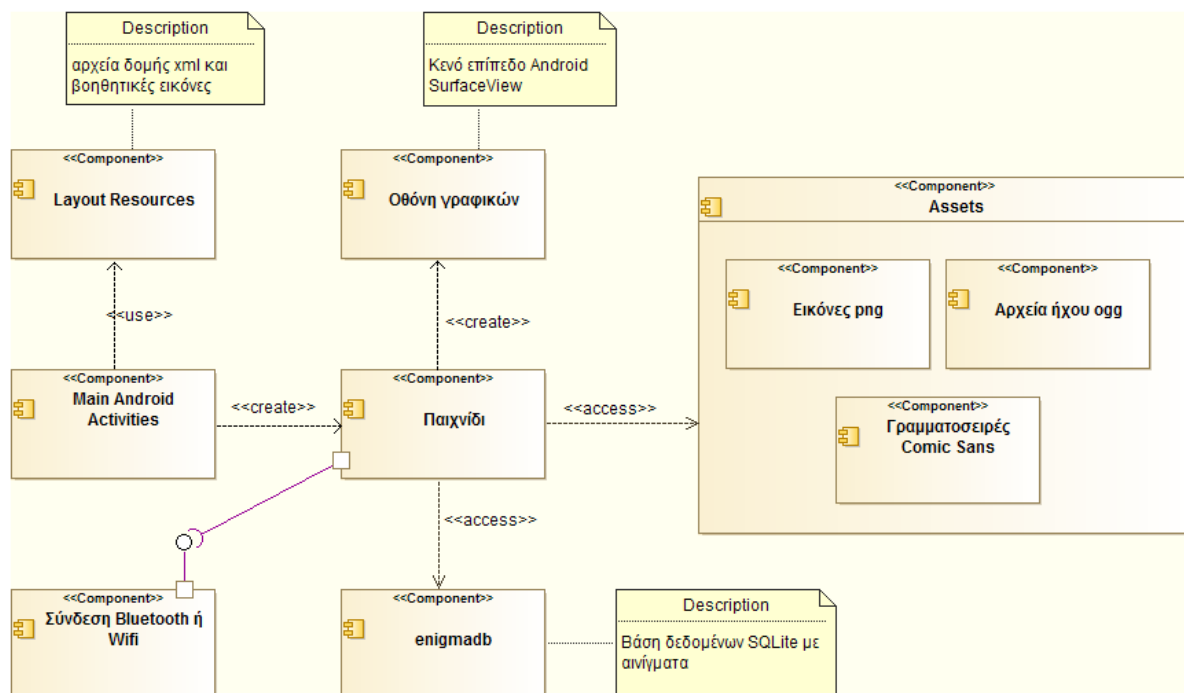


Σχήμα 2.16: UML Συγκεντρωτικό Διάγραμμα Κλάσεων.

2.4.5 Διάγραμμα Συστατικών (Component Diagram)

Χρησιμοποιείται για τη μοντελοποίηση των φυσικών πτυχών ενός συστήματος. Τα διαγράμματα συστατικών οπτικοποιούν την οργάνωση και τις σχέσεις μεταξύ κομματιών λογισμικού του

συστήματος. Είναι πολύ χρήσιμο από την πλευρά της υλοποίησης. Παρακάτω απεικονίζεται το αντίστοιχο διάγραμμα για την εφαρμογή Λαβύρινθος:



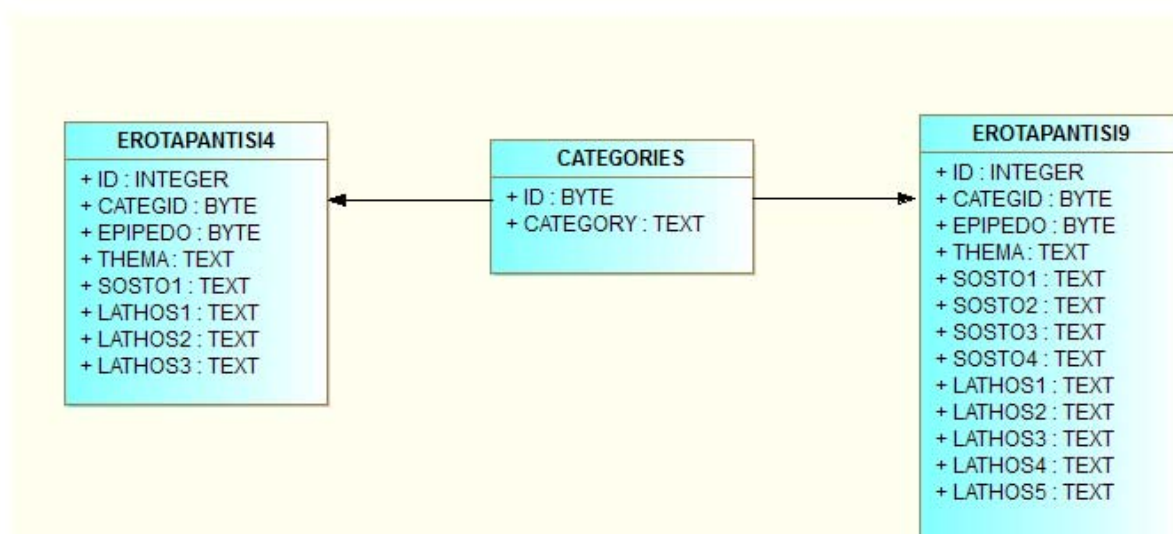
Σχήμα 2.17: UML Διάγραμμα Συστατικών.

2.4.6 Η Βάση Δεδομένων της Εφαρμογής

Το παιχνίδι χρειάζεται να αντλεί από κάπου τα αινίγματα (ερωτήσεις/απαντήσεις) για τις σφίγγες. Η δημιουργία μιας βάσης δεδομένων *enigmadb* μπορεί να αποτελέσει το αποθετήριο, από όπου θα αντλείται όλο το μαθησιακό υλικό. Το android διαθέτει εργαλεία για τη δημιουργία, τη σύνδεση και τη διαχείριση βάσεων δεδομένων τύπου SQLite. Η χρήση μιας τέτοιας βάσης δεδομένων από το παιχνίδι φαίνεται στο Διάγραμμα Συστατικών στο Σχήμα 2.17 (*enigmadb*).

Η δομή της βάσης δεδομένων μπορεί να είναι απλή. Τρεις πίνακες είναι αρκετοί, ώστε να κρατήσουν τα είδη των κατηγοριών, τις ερωτήσεις-απαντήσεις τύπου πολλαπλών επιλογών και τις ερωτήσεις-απαντήσεις τύπου πίνακα, αντίστοιχα. Στον πρώτο πίνακα της βάσης με όνομα *CATEGORIES* θα εισαχθούν οι κατηγορίες *Λογισμικό*, *Υλικό*, *Λειτουργικά Συστήματα*, *Διαδίκτυο*, *Σύστημα*, *Ερώτηση Κρίσης*. Στο δεύτερο πίνακα με όνομα *EROTAPANTISIA4* θα εισαχθούν οι ερωτήσεις, οι σωστές και λάθος απαντήσεις, η κατηγορία ερώτησης που αντιστοιχούν, και το επίπεδο δυσκολίας. Κάθε εγγραφή του πίνακα *EROTAPANTISIA4* θα περιλαμβάνει μία κατηγορία ερώτησης, το επίπεδο δυσκολίας που αντιστοιχεί, μία ερώτηση και τέσσερις απαντήσεις, εκ των

οποίων για την πρώτη κάνουμε την παραδοχή ότι θα αναγνωρίζεται πάντα ως η σωστή απάντηση, ενώ οι τρεις επόμενες θα είναι οι λάθος απαντήσεις. Παρόμοια σχεδιάζεται και το σχήμα του τρίτου πίνακα *EROTAPANTISI9*, όπου αντί για τέσσερα πεδία απαντήσεων θα πρέπει να δημιουργηθούν εννιά, εκ των οποίων οι απαντήσεις στα πρώτα τέσσερα θα αναγνωρίζονται από το σύστημα ως οι σωστές, ενώ οι επόμενες πέντε ως λάθος. Για όλους πίνακες θα υπάρχει ένα επιπλέον πεδίο *ID* ως μοναδικό πρωτεύον κλειδί που θα χαρακτηρίζει κάθε εγγραφή, ενώ για τους πίνακες *EROTAPANTISI4*, *EROTAPANTISI9*, ένα επιπρόσθετο ξένο κλειδί *CATEGID* θα αντιπροσωπεύει την κατηγορία που αντιστοιχεί στο *ID* του πίνακα *CATEGORIES*. Στο παρακάτω διάγραμμα αναπαρίσταται αναλυτικά η δομή της βάσης *enigmadb*.



Εικόνα 2.18: Η Βάση Δεδομένων *enigmadb*.

Έστω ότι οι ακέραιοι αριθμοί 1 έως 5 αντιστοιχούν στις 5 κατηγορίες ερωτήσεων και ο βαθμός δυσκολίας τους χαρακτηρίζεται από την τιμή του πεδίου *EPIPEDO*, ξεκινώντας κλιμακωτά από το 0 (μηδέν). Παρακάτω απεικονίζεται δείγμα εγγραφών από τον πίνακα *EROTAPANTISI4*.

<i>ID</i>	<i>CATEG ID</i>	<i>EPIPEDO</i>	<i>THEMA</i>	<i>SOSTO1</i>	<i>LATHOS1</i>	<i>LATHOS2</i>	<i>LATHOS3</i>
1	1	0	Τι είδους πρόγραμμα είναι η Ζωγραφική;	Σχεδιαστικό	Επεξεργασία ήχου	Πολυμέσα	Επεξεργασία κειμένου
2	1	0	Με ποιο πρόγραμμα μπορούμε να φτιάξουμε εύκολα παρουσιάσεις;	Powerpoint	Word	Excel	Outlook
3	4	0	Ποιος από τους παρακάτω είναι δυαδικός αριθμός;	11101	12345	22222	10002
4	2	1	Ποιο καλώδιο επιτρέπει ταχύτερη μεταφορά δεδομένων;	USB	Σειριακό καλώδιο	Παράλληλο LPT	PS/2

Πίνακας 2.3: Δείγμα έγκυρων τιμών για τα πεδία του πίνακα *EROTAPANTISI4*.

Η εφαρμογή *Λαβύρινθος* πρόκειται να έχει δύο επίπεδα δυσκολίας. Σύμφωνα με τις απαιτήσεις, οι ερωτήσεις που υποβάλλονται από τις σφίγγες στο δεύτερο λαβύρινθο είναι δυσκολότερες από αυτές του πρώτου. Επομένως, για τις εγγραφές με ερωτήσεις-απαντήσεις που θέλουμε να χαρακτηρίσουμε ως εύκολες καταχωρούμε στο πεδίο *EPIPEDO* την τιμή 0 (μηδέν), ενώ αντίστοιχα για τις πιο δύσκολες εισάγουμε την τιμή 1 (ένα).

Τέλος, δημιουργώντας στη βάση SQLite ευρετήρια για τα πεδία *CATEGID*, *EPIPEDO* των πινάκων *EROTAPANTISI4* και *EROTAPANTISI9* εξασφαλίζουμε ταχύτερες αναζητήσεις ερωτήσεων-απαντήσεων για συγκεκριμένη κατηγορία και επίπεδο.

Κεφάλαιο 3

Η Υλοποίηση της Εφαρμογής «Λαβύρινθος»

Στο κεφάλαιο αυτό παρουσιάζεται συνοπτικά η εφαρμογή «Λαβύρινθος» μετά την υλοποίησή της [18, 30-32]. Ως στοιχεία του παιχνιδιού χρησιμοποιούνται γνώριμες εικόνες και ήχοι που έχουν ξανασυναντηθεί σε πολλά άλλα δημοφιλή παιχνίδια, έτσι ώστε η εξοικείωση του μαθητή - χρήστη να είναι εύκολη και γρήγορη. Ένα παράδειγμα που συνηγορεί στο τελευταίο, είναι οι εικόνες που αντιπροσωπεύουν τις διαθέσιμες προσπάθειες/ζωές, που σκιαγραφούνται από κόκκινες καρδιές. Η εφαρμογή λειτουργεί στα ελληνικά.

Το δικτυακό μέρος του παιχνιδιού έχει υλοποιηθεί έως ένα σημείο, χωρίς ωστόσο να έχει ολοκληρωθεί. Εφόσον δύο παίκτες επιλέξουν ομαδικό παιχνίδι, δημιουργείται ασύρματη σύζευξη μεταξύ των συσκευών τους μέσω των bluetooth τους. Για να επιτευχθεί αυτό, οι συσκευές ξεκινάνε και οι δύο σαν εξυπηρετητές. Μόνο ο ένας παίκτης απαιτείται να επιλέξει να συνδεθεί στον άλλο. Ο δεύτερος απλά επιλέγει να συνεχίσει στην επόμενη οθόνη, όπως θα έκανε και στο ατομικό παιχνίδι και διατηρεί το ρόλο του εξυπηρετητή, ενώ ο προηγούμενος μετατρέπεται σε πελάτη (client). Έπειτα, η εφαρμογή στη συσκευή του εξυπηρετητή δημιουργεί έναν τυχαίο

λαβύρινθο με τα στοιχεία του και αμέσως μετά τα αποστέλλει στη συσκευή-πελάτη. Η εφαρμογή στη συσκευή-πελάτη διαβάζει τα δεδομένα που λήφθηκαν, τα ανακατασκευάζει και δημιουργεί έναν λαβύρινθο πανομοιότυπο με αυτόν της συσκευής-εξυπηρετητή. Έτσι, οι παίκτες εισέρχονται σε έναν κοινό λαβύρινθο με ίδιες θέσεις σφιγγών και καρδιών, με ίδια αινίγματα και για τους δύο. Ωστόσο από εκεί κι έπειτα ο καθένας συνεχίζει ατομικά και ό,τι αλληλεπίδραση έχει με το περιβάλλον του παιχνιδιού στη συσκευή του, παύει να έχει επίδραση στον άλλο.

3.1 Περιβάλλον ανάπτυξης και λειτουργίας

Το παιχνίδι αναπτύχθηκε με βάση την ανάλυση που έγινε στο Κεφάλαιο 2, στη γλώσσα Java σε συνδυασμό με το Android SDK, που περιλαμβάνει τις απαραίτητες βιβλιοθήκες για την ανάπτυξη εφαρμογών Android.

3.1.1 Android SDK APIs

Το Android SDK προσφέρει τα δημόσια APIs (Διασυνδέσεις Προγραμματισμού Εφαρμογών), τα οποία είναι σύνολα από βοηθητικές κλάσεις Java [18]. Τα κυριότερα πακέτα API που χρησιμοποιήθηκαν είναι τα εξής:

- [android.view](#) για το βασικό χειρισμό της διάταξης στοιχείων στην οθόνη και την αλληλεπίδρασή τους με το χρήστη (touchscreen).
- [android.graphics](#) για τη σχεδίαση και απόδοση των γραφικών, καθώς και την αποκωδικοποίηση και επεξεργασία των αρχείων εικόνας png που χρησιμοποιήθηκαν.
- [android.media](#) για την αποκωδικοποίηση και διαχείριση των αρχείων ήχου ogg.
- [android.database.sqlite](#) για τη δημιουργία και διαχείριση της ιδιωτικής βάσης δεδομένων SQLite.
- [android.bluetooth](#) για τη διαχείριση των λειτουργιών Bluetooth, όπως η σάρωση για συσκευές, η σύνδεση με συσκευές και η διαχείριση μεταφοράς δεδομένων μεταξύ των συσκευών.

- android.hardware για την υποστήριξη και διαχείριση του επιταχυνσιομέτρου (accelerometer).

Τα αρχεία εικόνας, ήχου, γραμματοσειράς και βάσης δεδομένων τοποθετούνται όλα στον προκαθορισμένο φάκελο *Assets*.

3.1.2 Εργαλεία και Περιβάλλον Ανάπτυξης

Η εφαρμογή «Λαβύρινθος» αναπτύχθηκε σε ένα σταθερό Η/Υ με λειτουργικό σύστημα Windows XP, ένα σταθερό Η/Υ με Windows 7 64bit και ένα φορητό Η/Υ με Windows 7 64bit. Τα εργαλεία/λογισμικά που απαιτήθηκαν να εγκατασταθούν και να χρησιμοποιηθούν για την ανάπτυξη της εφαρμογής είναι τα εξής:

- JDK (Java Development Kit) [19]. Όπως έχει προαναφερθεί, οι εφαρμογές για το λειτουργικό σύστημα Android γράφονται στη γλώσσα προγραμματισμού Java. Απαιτείται για την ανάπτυξη εφαρμογών Java.
- Modelio [20]. Εργαλείο σχεδίασης διαγραμμάτων UML (UML tool). Χρησιμοποιήθηκε για το σχεδιασμό του συστήματος και την παραγωγή μικρού μέρους κώδικα Java σε επίπεδο αρχιτεκτονικής των κλάσεων.
- Android Studio [21]. Περιβάλλον ανάπτυξης (IDE) της Google για τη γλώσσα Java για Android. Περιλαμβάνει το Android SDK και το ADT (Android Development Tools). Το ADT με τη σειρά του περιλαμβάνει εξομοιωτή πλατφόρμας Android για δοκιμές. Το σύνολο του κώδικα και η αποσφαλμάτωση της εφαρμογής «Λαβύρινθος» έγινε σε αυτό το περιβάλλον.
- VirtualBox [22]. Εικονική μηχανή (Εξομοιωτής υπολογιστικών συστημάτων). Χρησιμοποιήθηκε για τη δοκιμή του παιχνιδιού σε σταθερό Η/Υ με λειτουργικό σύστημα Windows, και τη λήψη στιγμιότυπων οθόνης κατά την εκτέλεσή του.
- Paint.NET [23]. «Ελαφρύ» λογισμικό επεξεργασίας εικόνας με περισσότερες δυνατότητες από το Paint των Windows. Χρησιμοποιήθηκε για την κατάλληλη τροποποίηση και μετατροπή σε μορφή png των αρχείων εικόνας της εφαρμογής.

- ButtonBass [24]. Online εργαλείο για τη δημιουργία απλών συνθέσεων μουσικής. Με τη βοήθειά του δημιουργήθηκε ένα μικρό μουσικό μοτίβο για την εφαρμογή.
- Audacity [25]. Λογισμικό επεξεργασίας ήχου. Χρησιμοποιήθηκε για τη δημιουργία και κατάλληλη επεξεργασία των μουσικών αρχείων ogg.
- DB Browser for SQLite [26]. Εργαλείο προβολής και διαχείρισης βάσεων δεδομένων SQLite. Με αυτό το εργαλείο έγινε η εισαγωγή των δεδομένων των αιτιγμάτων των σφιγγών στη βάση «enigmadb».
- Dropbox [27]. Εργαλείο με λειτουργίες Cloud Αποθήκευσης και συγχρονισμού. Χρησιμοποιήθηκε για την εύκολη πρόσβαση στην αναπτυσσόμενη εφαρμογή από όλους τους διαθέσιμους ηλεκτρονικούς υπολογιστές.

Όλα τα παραπάνω εργαλεία διατίθενται δωρεάν.

3.1.3 Απαιτήσεις συστήματος

Η εφαρμογή «Λαβύρινθος» εκτελείται πάνω σε οποιαδήποτε πλατφόρμα Android, είτε αυτή είναι σε smartphone, είτε σε tablet, είτε σε εξομοιωτή (emulator) σε PC. Αναλυτικά οι ελάχιστες απαιτήσεις συστήματος είναι:

- Πλατφόρμα με λειτουργικό σύστημα Android 2.3 Gingerbread (API 10) ή νεότερο.
- Bluetooth για το ομαδικό παιχνίδι.

3.2 Η διεπαφή συστήματος

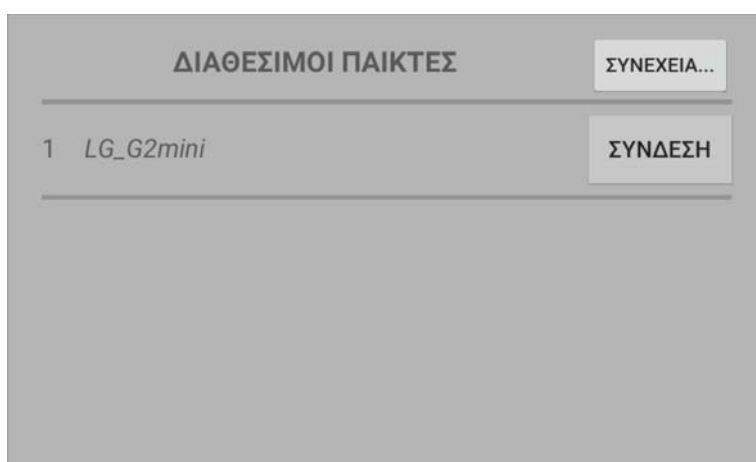
Καθώς ξεκινάει η εφαρμογή εμφανίζεται η εισαγωγική οθόνη καλωσορίσματος και το αρχικό μενού, τα οποία είναι το κύριο Android Activity.



Εικόνα 3.1: Εισαγωγική οθόνη καλωσορίσματος.

Η εικόνα που παρουσιάζεται ανατρέχει στο μύθο του Οιδίποδα και της Σφίγγας που του υποβάλλει το αίνιγμα, ως μια εισαγωγή στη λογική του παιχνιδιού. Δίνονται δύο επιλογές, όπου ο παίκτης μπορεί να ξεκινήσει το παιχνίδι, είτε μόνος του επιλέγοντας «ΑΤΟΜΙΚΟ ΠΑΙΧΝΙΔΙ», είτε σε συνεργασία με κάποιον άλλον που είναι κοντά του επιλέγοντας «ΟΜΑΔΙΚΟ ΠΑΙΧΝΙΔΙ».

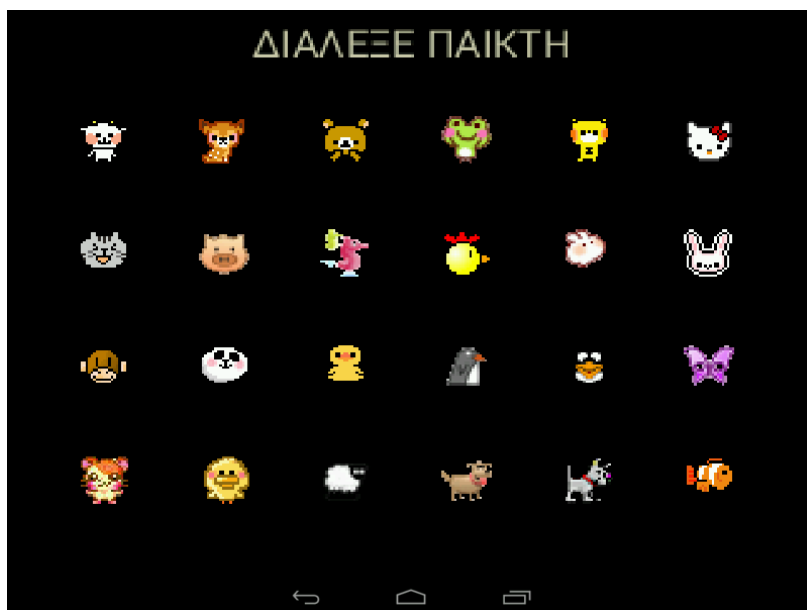
Αν ο χρήστης αποφασίσει να επιλέξει ομαδικό παιχνίδι, τότε ενεργοποιείται αυτόματα το Bluetooth της συσκευής του και εμφανίζεται επιπρόσθετα η παρακάτω οθόνη:



Εικόνα 3.2: Διεπαφή σύνδεσης με άλλους παίκτες.

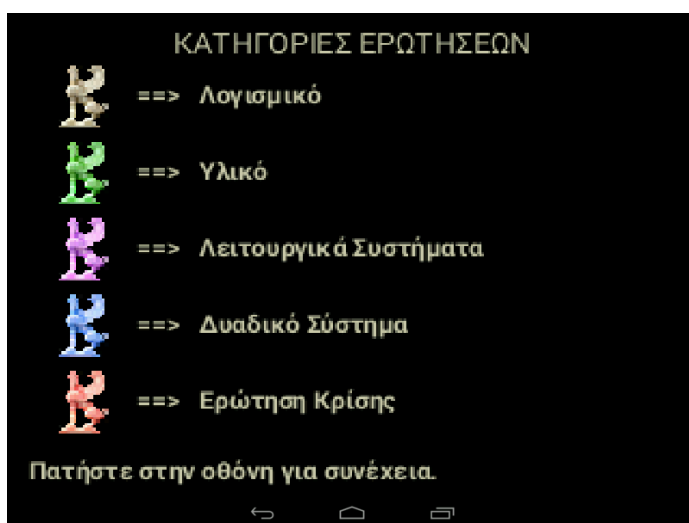
Στην οθόνη εμφανίζονται σε μορφή λίστας τυχόν υπάρχοντες παίκτες σε κοντινή εμβέλεια, που να έχουν κι εκείνοι ξεκινήσει την εφαρμογή Λαβύρινθος και ταυτόχρονα να έχουν επιλέξει επίσης ομαδικό παιχνίδι. Σε ένα ζευγάρι παικτών, ο ένας πατάει το κουμπί «ΣΥΝΔΕΣΗ» και συνδέεται στον άλλον δημιουργώντας ζεύγος, ενώ ο άλλος αντίθετα επιλέγει το κουμπί «ΣΥΝΕΧΕΙΑ...».

Αφού ο παίκτης επιλέξει έναν από τους δύο τρόπους παιχνιδιού (ατομικό ή ομαδικό), εμφανίζεται η παρακάτω οθόνη για την επιλογή εικόνας παίκτη (avatar):



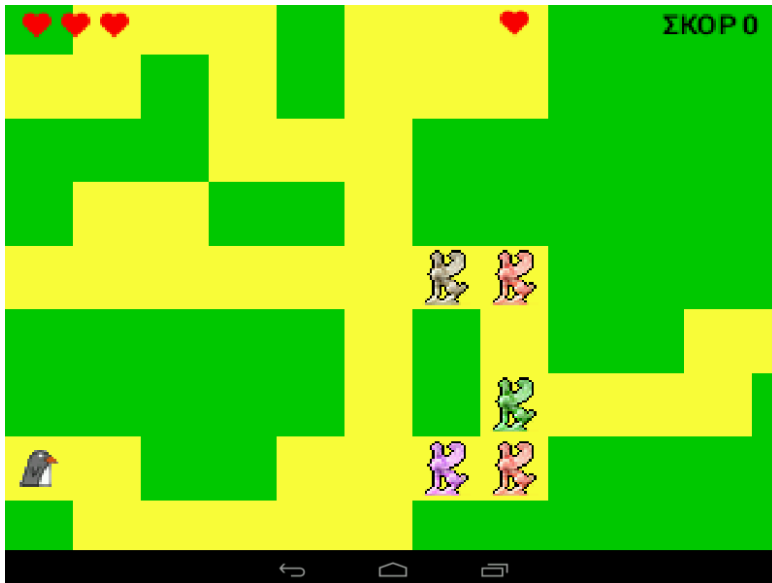
Εικόνα 3.3: Οθόνη επιλογής avatar.

Λίγο πριν ο παίκτης μπει στο λαβύρινθο με το avatar του, του δίνεται ένα ενημερωτικό υπόμνημα με τις σφίγγες και την αντιστοιχία τους χρώματός τους στην κατηγορία ερώτησης που κάνουν.



Εικόνα 3.4: Υπόμνημα με τις κατηγορίες των ερωτήσεων

Αφού ο παίκτης ακουμπήσει οπουδήποτε στην οθόνη, η εφαρμογή παράγει έναν τυχαίο λαβύρινθο, ο οποίος έχει πάντα επτά εισόδους και μία έξοδο. Οι θέσεις των σφιγγών, των καρδιών και της εξόδου είναι διαφορετικές σε κάθε εκτέλεση της εφαρμογής και παράγονται με τυχαίο τρόπο.



Εικόνα 3.5: Είσοδος στο Λαβύρινθο.

Ο παίκτης τοποθετείται τυχαία σε μια από τις εισόδους και μπορεί να ξεκινήσει να μετακινείται μέσα στον λαβύρινθο. Όταν ο παίκτης συναντήσει μία σφίγγα, αυτή του κάνει μία ερώτηση από συγκεκριμένη κατηγορία, ανάλογα με το χρώμα της, όπως έχει προαναφερθεί. Η παρακάτω εικόνα δείχνει μία ερώτηση τύπου πίνακα:



Εικόνα 3.6: Το αίνιγμα της Σφίγγας.

Κάθε λύση αινίγματος (εύρεση σωστών απαντήσεων) επιφέρει αύξηση του σκορ. Σκοπός του παίκτη είναι να ψάξει να βρει την έξοδο από το λαβύρινθο, η οποία απεικονίζεται με τη μορφή μιας πόρτας.

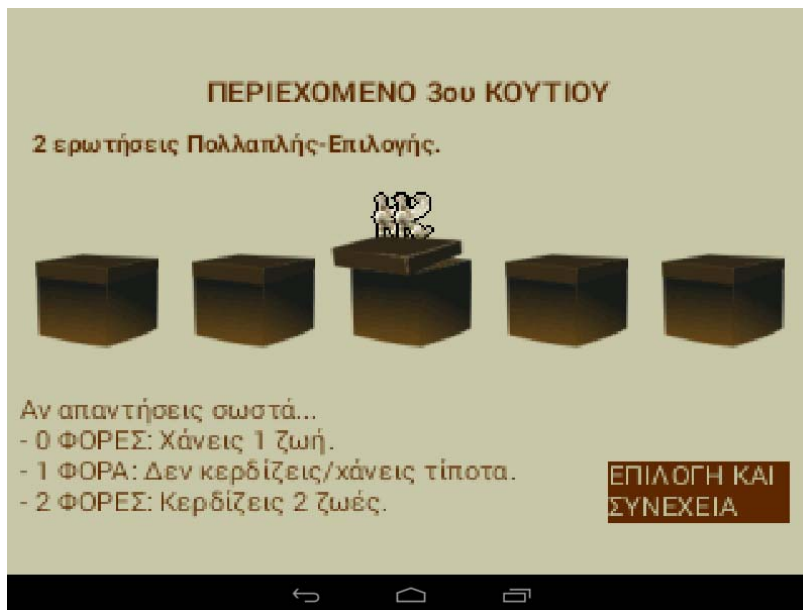


Εικόνα 3.7: Έξοδος από το λαβύρινθο εθεάθη.

Η πόρτα αρχικά είναι κλειστή και απαιτούνται οι λύσεις τουλάχιστον τριών αινιγμάτων από τις σφίγγες για να ξεκλειδώσει και να ανοίξει. Με αυτόν τον τρόπο αποφεύγονται οι περιπτώσεις, όπου η πόρτα έχει τοποθετηθεί σε πολύ εύκολο σημείο, με αποτέλεσμα ο παίκτης να μπορεί να αποφύγει τελείως τις ερωτήσεις.

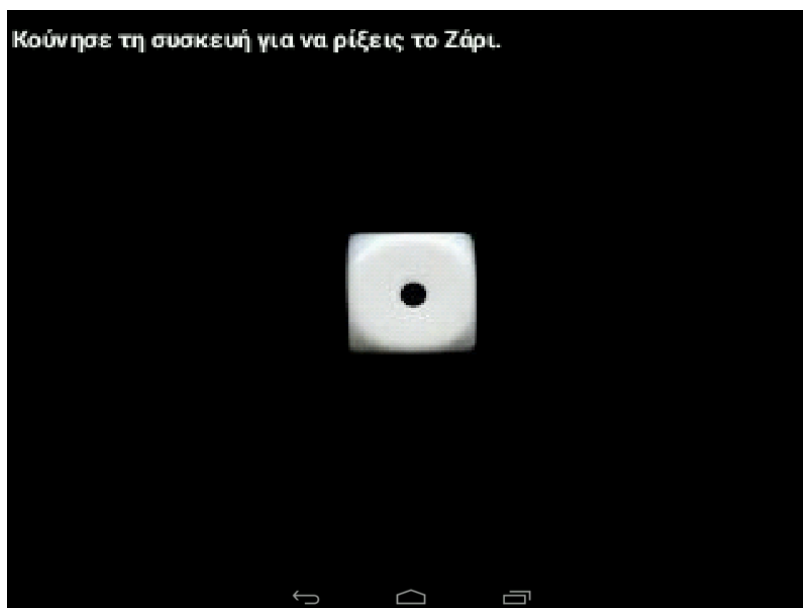
Μετά την επιτυχή έξοδο από τον λαβύρινθο ο παίκτης εισέρχεται σε μια μπόνους φάση, όπου μπορεί να κερδίσει ή να χάσει ζωές, χωρίς ωστόσο να μεταβάλλονται οι πόντοι του (σκορ). Εκεί, όπως αναφέρεται και στο κείμενο των ζητούμενων στο Κεφάλαιο 2, προβάλλονται 5 κουτιά, για το περιεχόμενο των οποίων δίνεται σύντομη περιγραφή, κατά το άγγιγμά τους.

Στην επόμενη εικόνα φαίνεται στιγμιότυπο της οθόνης μπόνους φάση:



Εικόνα 3.8: Μπόνους φάση.

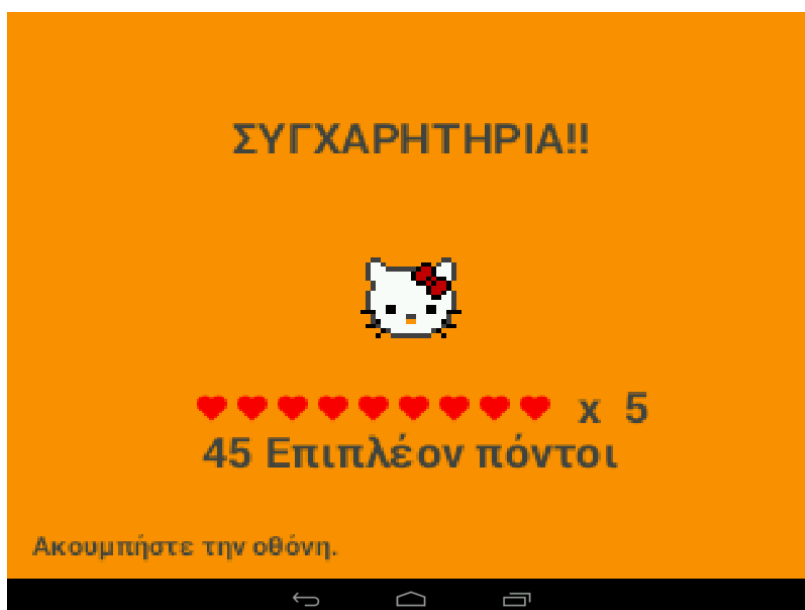
Ο παίκτης αν επιθυμεί να μη λύσει κάποιο επιπλέον αίνιγμα, αλλά παρόλα αυτά θέλει να δοκιμάσει την τύχη του, μπορεί να επιλέξει το πέμπτο κουτί, που περιλαμβάνει ένα ζάρι.



Εικόνα 3.9: Οθόνη με το Ζάρι.

Σε κάθε περίπτωση, ο παίκτης δε γίνεται να χάσει το παιχνίδι στη μπόνους φάση, αλλά ακολούθως συνεχίζει στον δεύτερο λαβύρινθο, όπου αυτήν τη φορά οι ερωτήσεις έχουν αυξημένο επίπεδο δυσκολίας σε σχέση με τον πρώτο.

Αν ο παίκτης καταφέρει να βγει από το λαβύρινθο και τη δεύτερη φορά, το παιχνίδι μετατρέπει τις περισσευούμενες ζωές (καρδιές) σε επιπλέον πόντους και τερματίζει, αναγράφοντας το τελικό του σκορ και σχολιάζοντας με ένα μήνυμα την απόδοσή του.



Εικόνα 3.10: Επιτυχής Τερματισμός.

3.3 Χαρακτηριστικά και Αξιοποιούμενες Δυνατότητες

Η εφαρμογή Λαβύρινθος αξιοποιεί εξειδικευμένες δυνατότητες που παρέχουν οι πλατφόρμες κινητού υπολογισμού όπως:

1. Δυνατότητες σύνδεσης με άλλες συσκευές «εν κινήσει», μέσω του Bluetooth, χωρίς την εξάρτηση από δρομολογητές και την ύπαρξη άλλων ασύρματων ή μη τοπικών δικτύων και internet.
2. Χειρισμός του παιχνιδιού με αγγίγματα στην οθόνη (touchscreen).
3. Χειρισμός του παιχνιδιού, μέσω της αλλαγής του προσανατολισμού της συσκευής (π.χ. για να προχωρήσει ο χαρακτήρας δεξιά στο λαβύρινθο στρίβουμε τη συσκευή δεξιά). Αυτό επιτυγχάνεται με την αξιοποίηση του αισθητήρα που διαθέτει η συντριπτική πλειοψηφία των έξυπνων κινητών συσκευών, το επιταχυνσιόμετρο (accelerometer).
4. Αίσθηση πραγματικού ριξίματος ζαριού κουνώντας τη συσκευή, μέσω της χρήσης του επιταχυνσιομέτρου.

5. Δόνηση της συσκευής κατά την αποτυχία επίλυσης αινίγματος.
6. Χρήση πολυμέσων. Περιλαμβάνονται εικόνες σε συνδυασμό με ήχους και μουσική, αλλά και αναπαραγωγή κομματιών βίντεο.

Κεφάλαιο 4

Αποτίμηση και Αξιολόγηση

Η σχεδίαση και το ψυχαγωγικό και εκπαιδευτικό υλικό, που χρησιμοποιήθηκε στο εκπαιδευτικό παιχνίδι Λαβύρινθος απευθύνεται κυρίως σε παιδιά πέμπτης και έκτης δημοτικού σχολείου. Οι ερωτήσεις και απαντήσεις που τοποθετήθηκαν στην παρεχόμενη βάση δεδομένων αφορούν το μάθημα της Πληροφορικής δημοτικού.

Για την αποτίμηση και αξιολόγηση του λογισμικού ζητήθηκαν οι απόψεις έντεκα εν ενεργεία εκπαιδευτικών από διαφορετικές περιοχές της Ελλάδας, που διδάσκουν ή έχουν διδάξει σε δημοτικά σχολεία, εκ των οποίων οι περισσότεροι είναι του κλάδου της πληροφορικής. Ωστόσο οι τρεις εξ αυτών είναι εκπαιδευτικοί άλλων κλάδων, με την πεποίθηση ότι θα συμβάλλουν στην απόκτηση μιας πιο σφαιρικής εικόνας.

Το εύρος ηλικιών των εκπαιδευτικών που συμμετείχαν στην έρευνα κυμαίνεται μεταξύ 30 – 39 έτη. Οι ειδικότητές τους παρατίθενται στον παρακάτω πίνακα:

ΕΙΔΙΚΟΤΗΤΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ	ΠΛΗΘΟΣ
ΠΕ06 ΑΓΓΛΙΚΗΣ ΦΙΛΟΛΟΓΙΑΣ	1
ΠΕ04.01 ΦΥΣΙΚΟΙ	1
ΠΕ14.04 ΓΕΩΠΟΝΟΙ	1
ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ ΑΕΙ	5
ΠΕ20 ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕΙ	3

Πίνακας 4.1: Κλάδοι Συμμετεχόντων Εκπαιδευτικών

Η συλλογή των απόψεων των εκπαιδευτικών έγινε με τη μορφή ερωτηματολογίων κλειστού τύπου, ώστε να είναι ευκολότερη η διαδικασία, με την εξαίρεση μια τελικής ανοιχτής ερώτησης. Τα ερωτηματολόγια χωρίζονται σε δύο κατηγορίες. Το ερωτηματολόγιο πριν την υλοποίηση και το ερωτηματολόγιο μετά την υλοποίηση. Η καταγραφή των απαντήσεων των εκπαιδευτικών γίνεται ανά ειδικότητα.

4.1 Ανάλυση αποτελεσμάτων ερωτηματολογίου εκπαιδευτικών πριν την υλοποίηση.

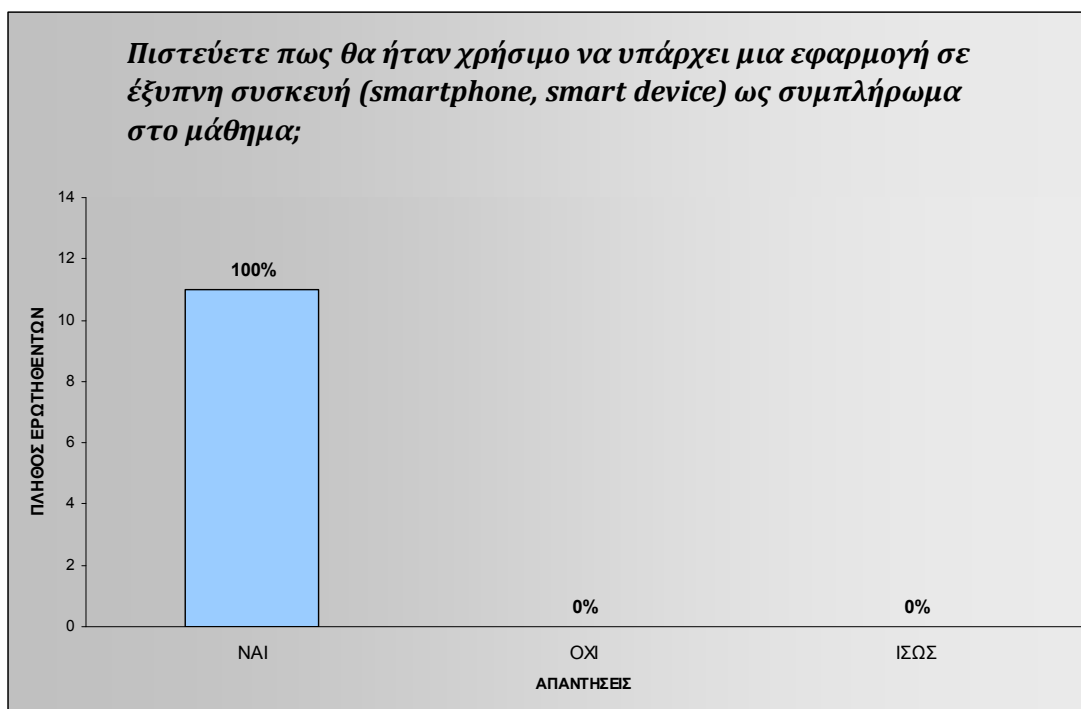
Πριν προχωρήσουμε στην υλοποίηση του εκπαιδευτικού παιχνιδιού Λαβύρινθος, συλλέξαμε τα ακόλουθα αποτελέσματα από τις απαντήσεις που λάβαμε από τους έντεκα εκπαιδευτικούς.

Ερώτηση 1: Πιστεύετε πως θα ήταν χρήσιμο να υπάρχει μια εφαρμογή σε έξυπνη συσκευή (smartphone, smart device) ως συμπλήρωμα στο μάθημα;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην πρώτη ερώτηση καταγράφονται στον Πίνακα 4.2 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.1:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Πίνακας 4.2: Απαντήσεις στην 1η ερώτηση πριν την υλοποίηση.



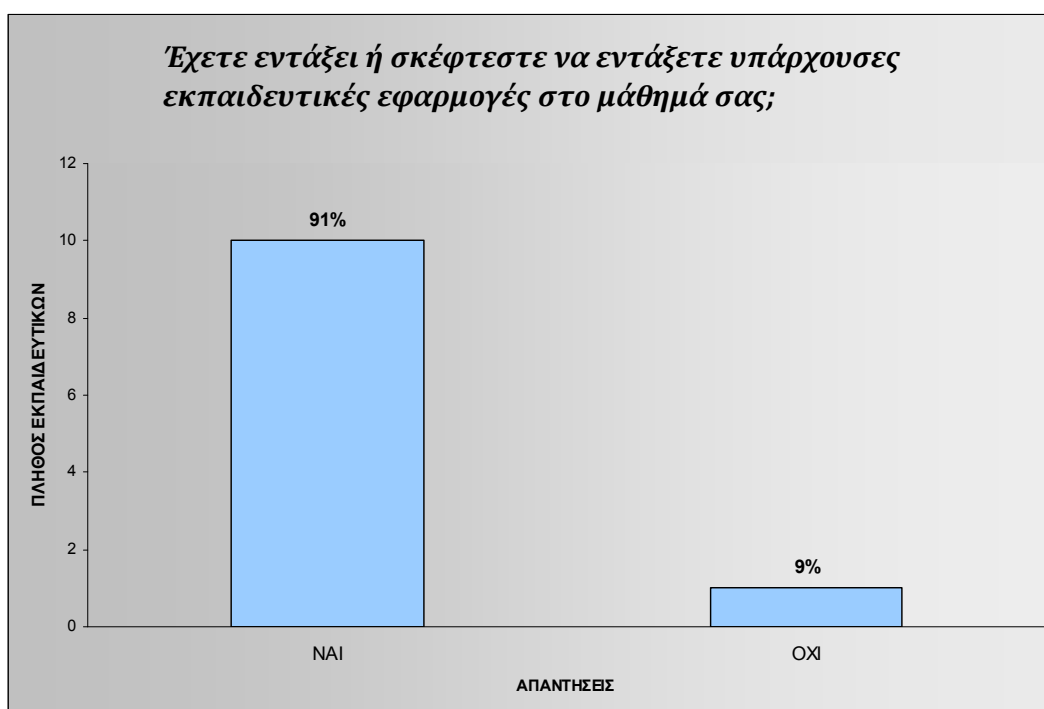
Γράφημα 4.1: Πόσοι πιστεύουν πως θα ήταν χρήσιμη μια εφαρμογή σε έξυπνη συσκευή ως συμπλήρωμα στο μάθημα.

Ερώτηση 2: Έχετε εντάξει ή σκέφτεστε να εντάξετε υπάρχουσες εκπαιδευτικές εφαρμογές στο μάθημά σας;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στη δεύτερη ερώτηση καταγράφονται στον Πίνακα 4.3 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.2:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Πίνακας 4.3: Απαντήσεις στη 2η ερώτηση πριν την υλοποίηση.



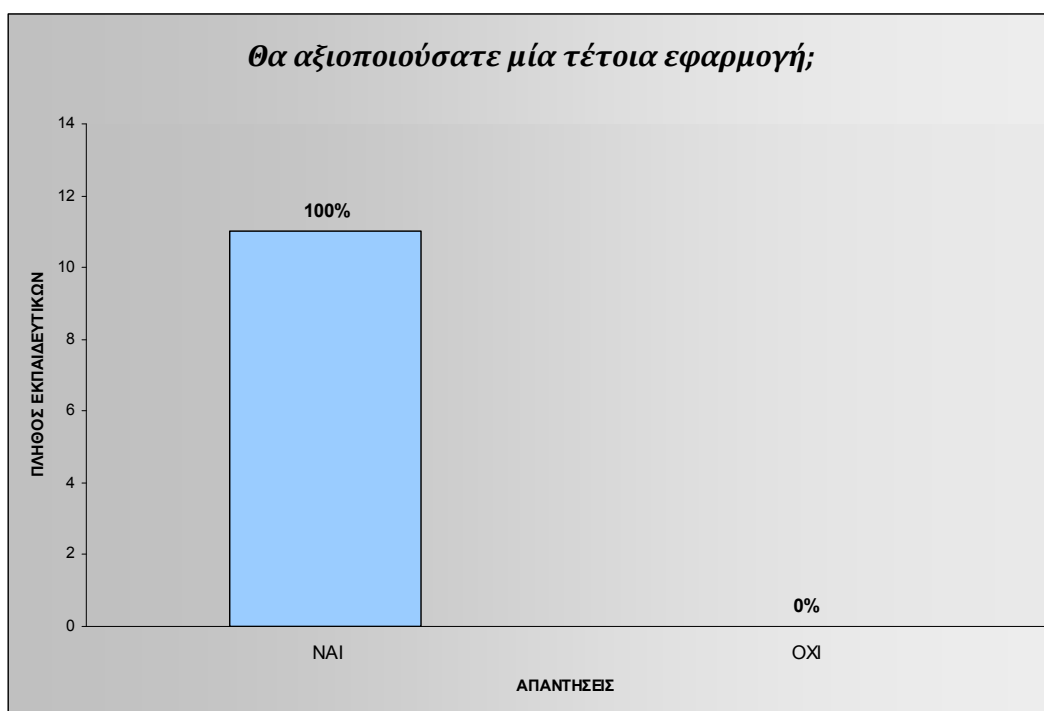
Γράφημα 4.2: Πόσοι έχουν εντάξει ή σκέφτονται να εντάξουν εκπαιδευτικές εφαρμογές στο μάθημα.

Ερώτηση 3: Θα αξιοποιούσατε μία τέτοια εφαρμογή;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην τρίτη ερώτηση καταγράφονται στον Πίνακα 4.4 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.3:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Πίνακας 4.4: Απαντήσεις στην 3η ερώτηση πριν την υλοποίηση.



Γράφημα 4.3: Πόσοι θα αξιοποιούσαν μια τέτοια εφαρμογή.

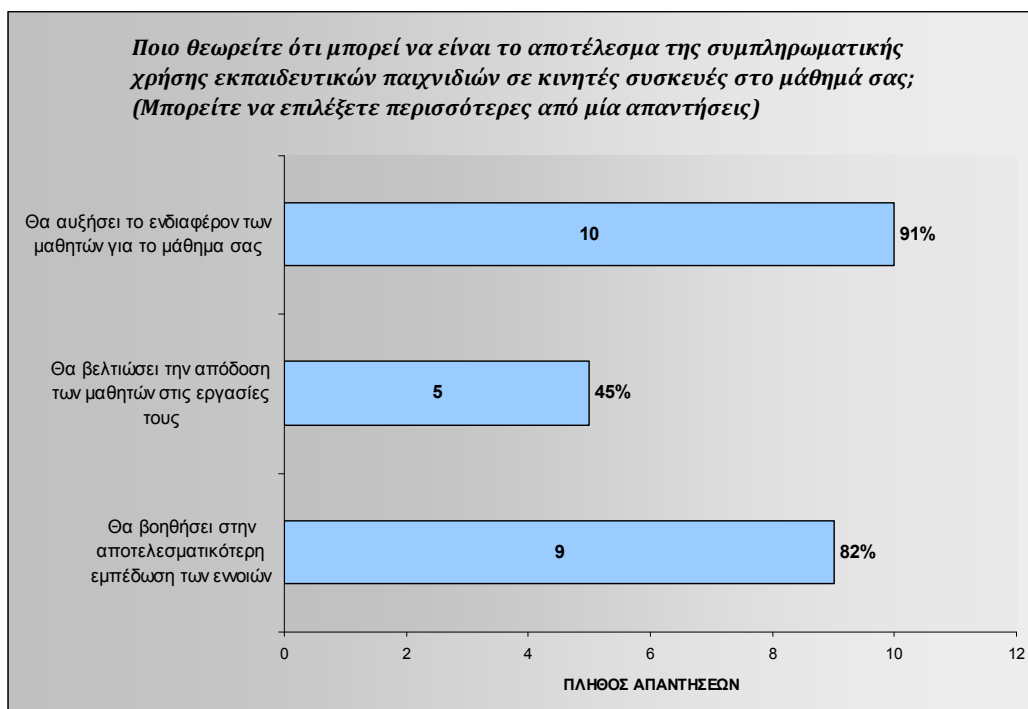
Ερώτηση 4: Ποιο θεωρείτε ότι μπορεί να είναι το αποτέλεσμα της συμπληρωματικής χρήσης εκπαιδευτικών παιχνιδιών σε κινητές συσκευές στο μάθημά σας; (Μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις)

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην τέταρτη ερώτηση καταγράφονται στον Πίνακα 4.5 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.4:

ΠΕ19 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας
ΠΕ19 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους

ΠΕ19 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ14.04 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ19 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ04.01 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ20 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ06 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ19 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ20 Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών
ΠΕ20 Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών

Πίνακας 4.5: Απαντήσεις στην 4η ερώτηση πριν την υλοποίηση.



Γράφημα 4.4: Ποιο θεωρείτε ότι μπορεί να είναι το αποτέλεσμα της συμπληρωματικής χρήσης εκπαιδευτικών παιχνιδιών σε κινητές συσκευές στο μάθημα.

4.2 Ερωτηματολόγιο εκπαιδευτικών μετά την υλοποίηση

Στη συνέχεια παρατίθενται οι απόψεις 9 εκπαιδευτικών, όπως καταγράφηκαν σε ερωτηματολόγια για την αποτύπωση των εντυπώσεών τους από τη χρήση του Λαβύρινθου. Οι ερωτήσεις εστίασαν στην άποψή τους για το Λαβύρινθο ως ψυχαγωγικό εργαλείο για τη διδασκαλία και τη μάθηση εννοιών πληροφορικής στις τελευταίες τάξεις του δημοτικού, τη δεκτικότητά τους όσον αφορά την αξιοποίηση των τεχνολογιών στις πλατφόρμες κινητού υπολογισμού, στον εντοπισμό κάποιου σημείου που τους προβληματίζει και στις προτάσεις τους για τυχόν βελτιώσεις.

Ερώτηση 1: Πόσο ενδιαφέρουσα βρήκατε την εφαρμογή σε κλίμακα 1 – 10;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην πρώτη ερώτηση καταγράφονται στον Πίνακα 4.6:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
8	9	7	5	10	10	10	7	8	9	10

Πίνακας 4.6: Απαντήσεις στην 1η ερώτηση μετά την υλοποίηση.

Ο μέσος όρος που προκύπτει από τις βαθμολογίες των απαντήσεων είναι **8,5**.

Ερώτηση 2: Πόσο εύχρηστη βρήκατε τη διεπαφή της εφαρμογής σε κλίμακα 1 – 10;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στη δεύτερη ερώτηση καταγράφονται στον Πίνακα 4.7:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
8	7	7	5	9	8	9	6	8	10	10

Πίνακας 4.7: Απαντήσεις στη 2η ερώτηση μετά την υλοποίηση.

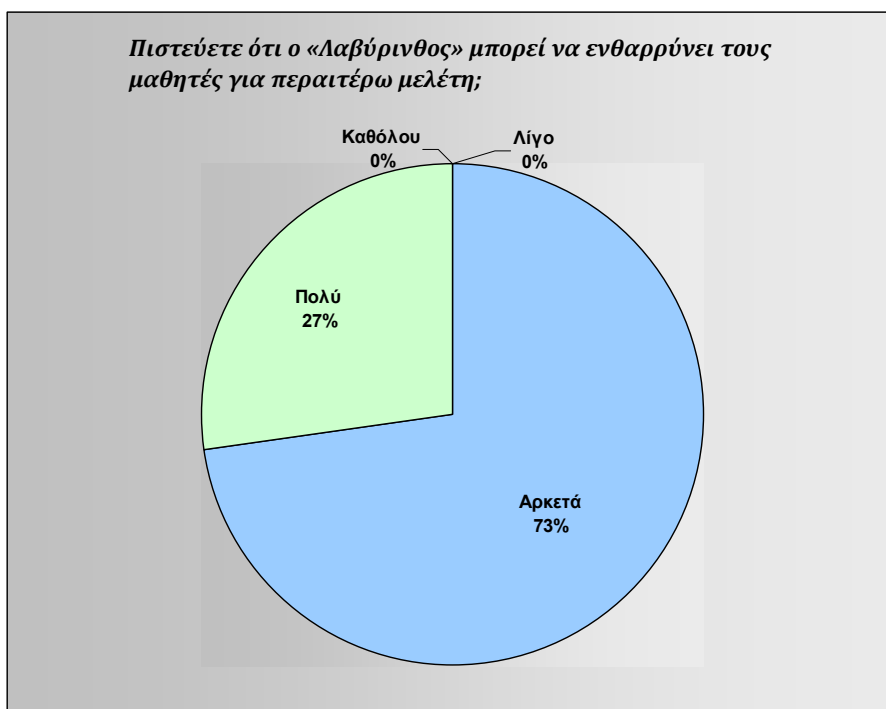
Ο μέσος όρος που προκύπτει από τις βαθμολογίες των απαντήσεων είναι **7,9**.

Ερώτηση 3: Πιστεύετε ότι ο «Λαβύρινθος» μπορεί να ενθαρρύνει τους μαθητές για περαιτέρω μελέτη;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην τρίτη ερώτηση καταγράφονται στον Πίνακα 4.8 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.5:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Αρκετά	Πολύ	Αρκετά	Αρκετά	Πολύ	Πολύ	Αρκετά	Αρκετά	Αρκετά	Αρκετά	Αρκετά

Πίνακας 4.8: Απαντήσεις στην 3η ερώτηση μετά την υλοποίηση.



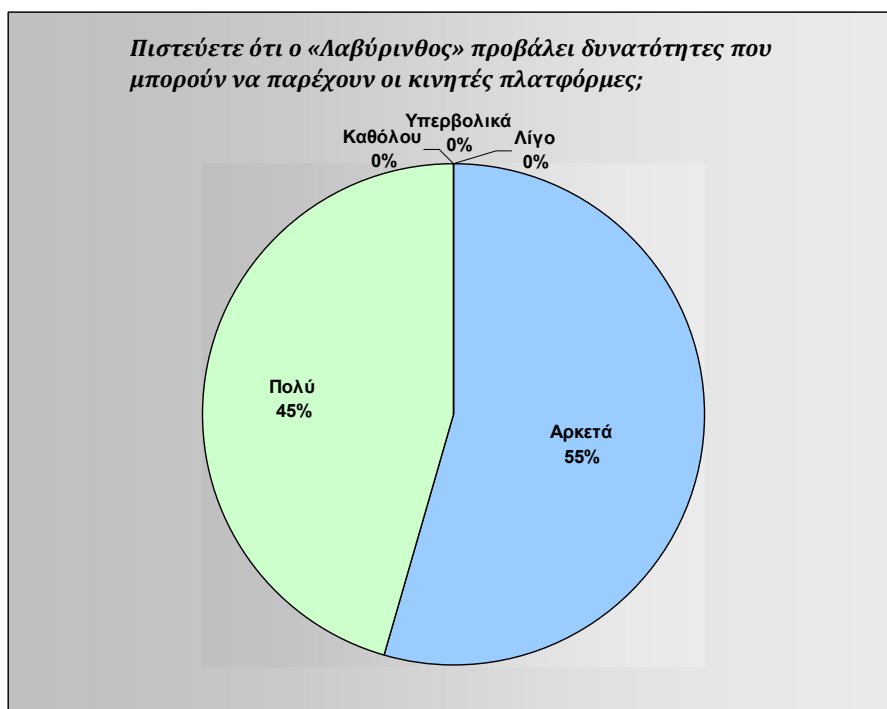
Γράφημα 4.5: Άποψη αν ο «Λαβύρινθος» μπορεί να ενθαρρύνει τους μαθητές για περαιτέρω μελέτη.

Ερώτηση 4: Πιστεύετε ότι ο «Λαβύρινθος» προβάλλει δυνατότητες που μπορούν να παρέχουν οι κινητές πλατφόρμες;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην τέταρτη ερώτηση καταγράφονται στον Πίνακα 4.9 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.6:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Αρκετά	Πολύ	Αρκετά	Αρκετά	Πολύ	Αρκετά	Πολύ	Αρκετά	Πολύ	Πολύ	Αρκετά

Πίνακας 4.9: Απαντήσεις στην 4η ερώτηση μετά την υλοποίηση.



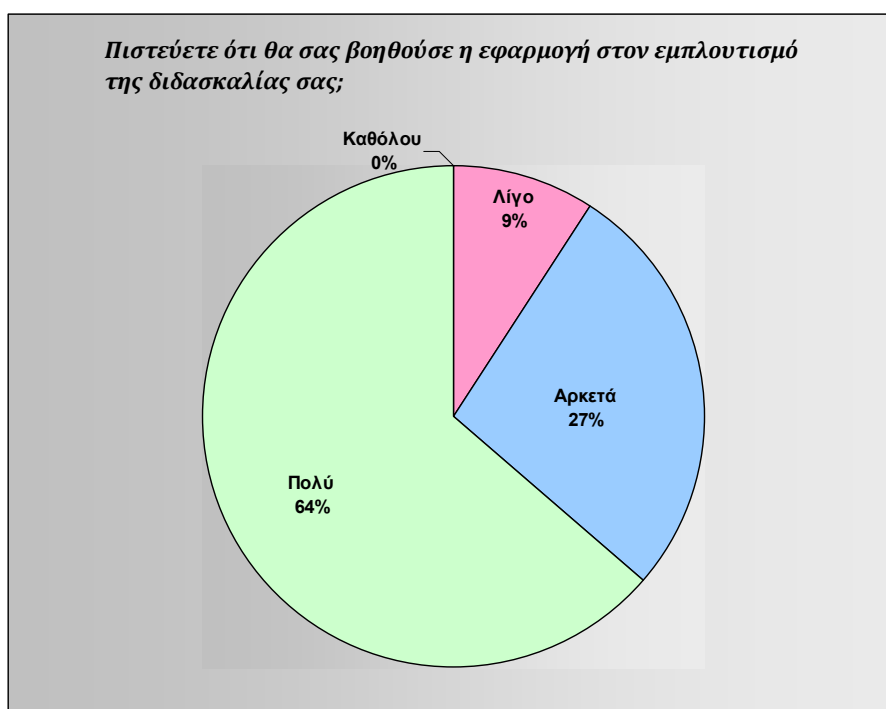
Γράφημα 4.6: Άποψη αν ο «Λαβύρινθος» προβάλλει δυνατότητες που παρέχουν οι κινητές πλατφόρμες.

Ερώτηση 5: Πιστεύετε ότι θα σας βοηθούσε η εφαρμογή στον εμπλουτισμό της διδασκαλίας σας;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην πέμπτη ερώτηση καταγράφονται στον Πίνακα 4.10 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.7:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Πολύ	Πολύ	Πολύ	Λίγο	Πολύ	Πολύ	Αρκετά	Αρκετά	Αρκετά	Πολύ	Πολύ

Πίνακας 4.10: Απαντήσεις στην 5η ερώτηση μετά την υλοποίηση.



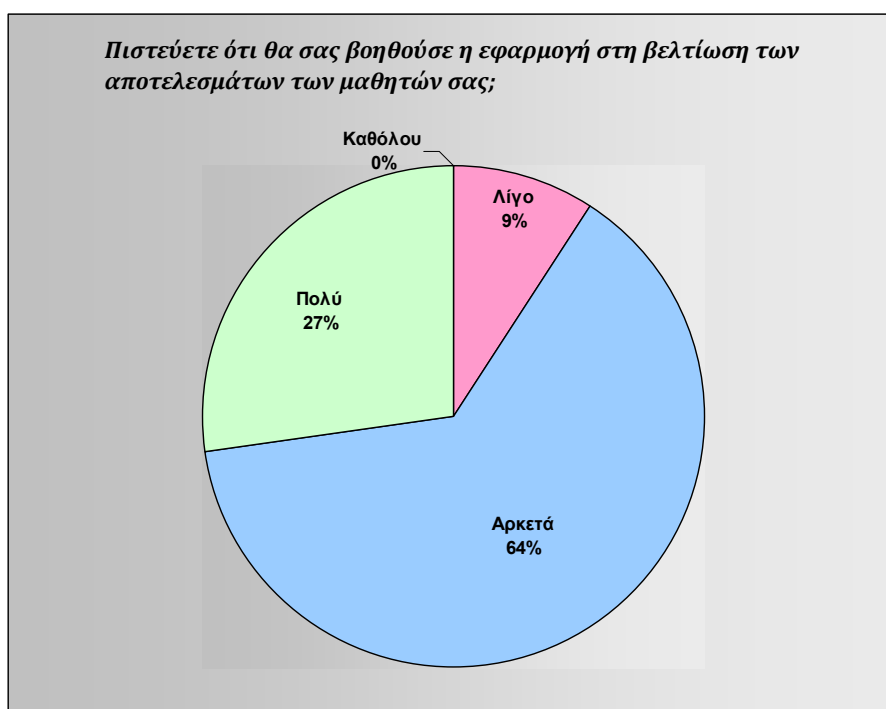
Γράφημα 4.7: Άποψη αν ο «Λαβύρινθος» θα βοηθούσε η εφαρμογή στον εμπλουτισμό της διδασκαλίας.

Ερώτηση 6: Πιστεύετε ότι θα σας βοηθούσε η εφαρμογή στη βελτίωση των αποτελεσμάτων των μαθητών σας;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην έκτη ερώτηση καταγράφονται στον Πίνακα 4.11 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.8:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Αρκετά	Πολύ	Αρκετά	Λίγο	Πολύ	Αρκετά	Αρκετά	Αρκετά	Αρκετά	Πολύ	Αρκετά

Πίνακας 4.11: Απαντήσεις στην 6η ερώτηση μετά την υλοποίηση.



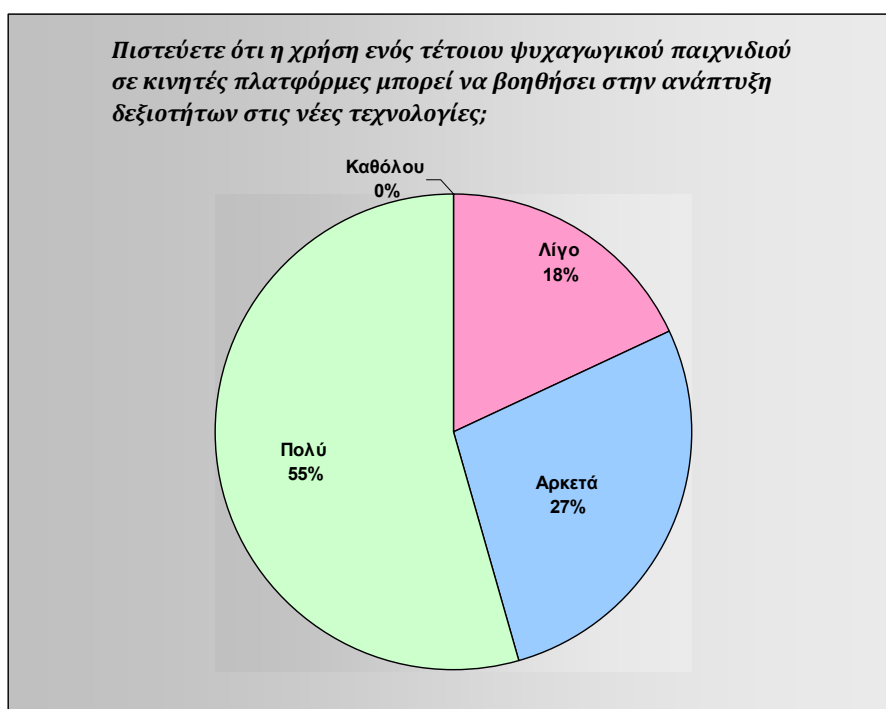
Γράφημα 4.8: Άποψη αν η εφαρμογή θα βοηθούσε στη βελτίωση των αποτελεσμάτων των μαθητών.

Ερώτηση 7: Πιστεύετε ότι η χρήση ενός τέτοιου ψυχαγωγικού παιχνιδιού σε κινητές πλατφόρμες μπορεί να βοηθήσει στην ανάπτυξη δεξιοτήτων στις νέες τεχνολογίες;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην έβδομη ερώτηση καταγράφονται στον Πίνακα 4.12 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.9:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Λίγο	Πολύ	Πολύ	Πολύ	Πολύ	Αρκετά	Αρκετά	Λίγο	Πολύ	Πολύ	Αρκετά

Πίνακας 4.12: Απαντήσεις στην 7η ερώτηση μετά την υλοποίηση.



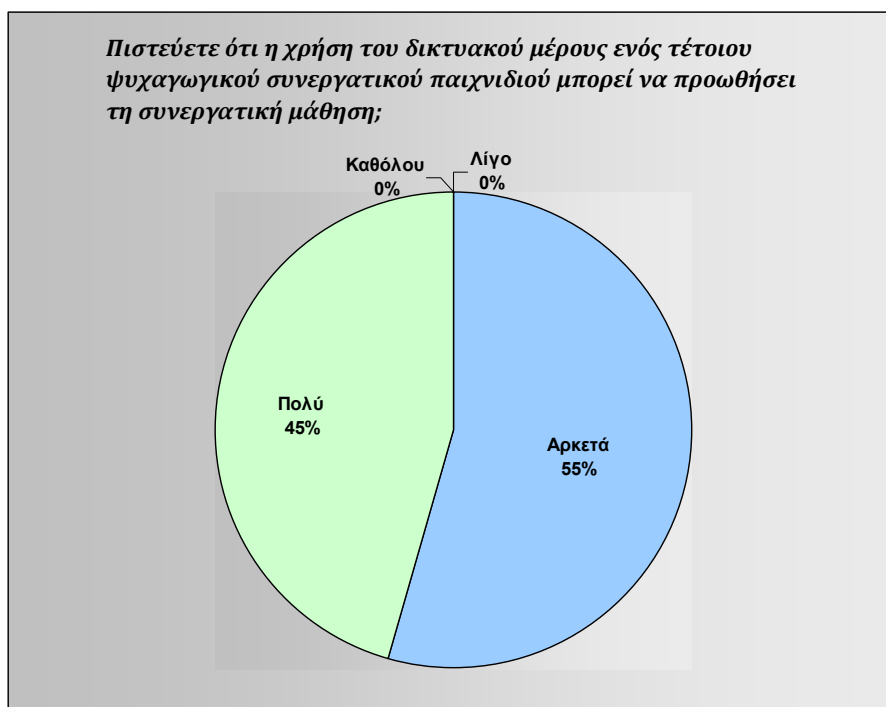
Γράφημα 4.9: Άποψη αν η χρήση ενός τέτοιου ψυχαγωγικού παιχνιδιού μπορεί να βοηθήσει στην ανάπτυξη δεξιοτήτων στις νέες τεχνολογίες.

Ερώτηση 8: Πιστεύετε ότι η χρήση του δικτυακού μέρους ενός τέτοιου ψυχαγωγικού συνεργατικού παιχνιδιού μπορεί να προωθήσει τη συνεργατική μάθηση;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην όγδοη ερώτηση καταγράφονται στον Πίνακα 4.13 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.10:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
Αρκετά	Πολύ	Πολύ	Αρκετά	Πολύ	Αρκετά	Αρκετά	Αρκετά	Αρκετά	Πολύ	Πολύ

Πίνακας 4.13: Απαντήσεις στην 8η ερώτηση μετά την υλοποίηση.



Γράφημα 4.10: Άποψη αν η χρήση του δικτυακού μέρους ενός τέτοιου ψυχαγωγικού συνεργατικού παιχνιδιού μπορεί να προωθήσει τη συνεργατική μάθηση.

Ερώτηση 9: Πόσο θα βαθμολογούσατε την εφαρμογή από διδακτικής άποψης σε κλίμακα 1 – 10;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην ένατη ερώτηση καταγράφονται στον Πίνακα 4.14:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
10	9	8	7	9	8	8	8	10	10	9

Πίνακας 4.14: Απαντήσεις στην 9η ερώτηση μετά την υλοποίηση.

Ο μέσος όρος που προκύπτει από τις βαθμολογίες των απαντήσεων είναι **8,7**.

Ερώτηση 10: Πόσο θα βαθμολογούσατε την εφαρμογή από ψυχαγωγικής άποψης σε κλίμακα 1 – 10;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στη δέκατη ερώτηση καταγράφονται στον Πίνακα 4.15:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
5	10	7	6	10	9	10	10	8	10	9

Πίνακας 4.15: Απαντήσεις στη 10η ερώτηση μετά την υλοποίηση.

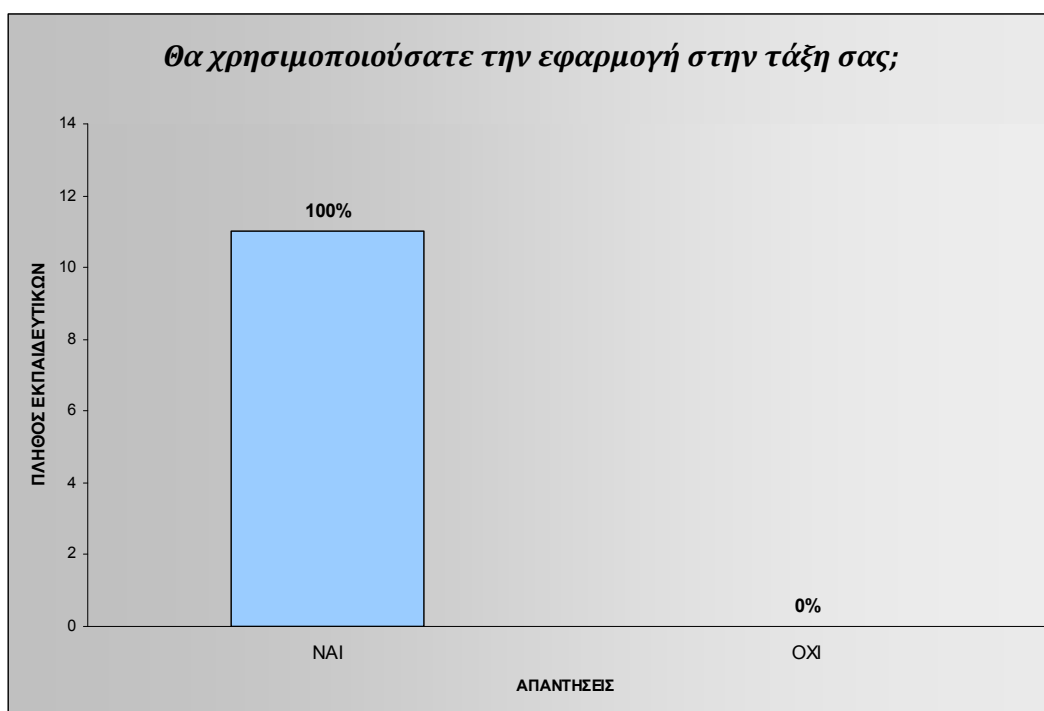
Ο μέσος όρος που προκύπτει από τις βαθμολογίες των απαντήσεων είναι **8,5**.

Ερώτηση 11: Θα χρησιμοποιούσατε την εφαρμογή στην τάξη σας;

Οι απαντήσεις που δόθηκαν από τους εκπαιδευτικούς στην ενδέκατη ερώτηση καταγράφονται στον Πίνακα 4.16 και η σύνοψη των απαντήσεων φαίνεται στο Γράφημα 4.11:

ΠΕ19	ΠΕ19	ΠΕ19	ΠΕ14.04	ΠΕ19	ΠΕ04.01	ΠΕ20	ΠΕ06	ΠΕ19	ΠΕ20	ΠΕ20
ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Πίνακας 4.16: Απαντήσεις στη 11η ερώτηση μετά την υλοποίηση.



Γράφημα 4.11: Χρήση της εφαρμογής στην τάξη.

Ερώτηση 12: Αν απαντήσατε «Όχι» στην προηγούμενη ερώτηση, παρακαλώ αιτιολογήστε.

Δεν υπήρξε κάποια αρνητική απάντηση από κανέναν εκπαιδευτικό.

4.3 Παρατηρήσεις και προτεινόμενες βελτιώσεις από εκπαιδευτικούς

Το τελευταίο στάδιο της αξιολόγησης της εφαρμογής Λαβύρινθος από τους συμμετέχοντες εκπαιδευτικούς περιλαμβάνει παρατηρήσεις, συμβουλές και προτάσεις για πιθανές βελτιώσεις. Οι προτάσεις που ακολουθούν αυτούσιες, όπως ακριβώς τις έχουν διατυπώσει.

Εκπαιδευτικός ΠΕ19: «Μεγαλύτερη ανάλυση ώστε να είναι πιο ευδιάκριτα τα γράμματα και τα γραφικά. Προσθήκη animation. Τα παιδιά έχουν συνηθίσει σε γρήγορη εναλλαγή εικόνων. Προσθήκη high-score σε ατομικό αλλά και ομαδικό παιχνίδι. Στα παιδιά αρέσει ο ανταγωνισμός».

Εκπαιδευτικός ΠΕ19: «Έχει μερικά bugs, όπως για παράδειγμα όταν ξεκινά το παιχνίδι, η εικόνα κατεβαίνει εκτός λαβυρίνθου και δεν μπορώ να παίξω το παιχνίδι, οπότε πρέπει να το ξαναρχινήσω. Επίσης στα κουμπιά καλό είναι να υπάρχει ένα πλαίσιο σε διαφορετικό χρώμα, ώστε να καταλαβαίνει κάποιος ότι είναι κουμπί. Συνήθως για λόγους ευχρηστίας επιλέγουμε συγκεκριμένα χρώματα που κάνουν μια «καλή» αντίθεση στο μάτι π.χ μαύρο-κίτρινο».

Εκπαιδευτικός ΠΕ19: «Οι ερωτήσεις είναι αρκετές, αλλά καλό είναι να μπουν ερωτήσεις και για άλλα μαθήματα».

Εκπαιδευτικός ΠΕ14.04: «Το επίπεδο των ερωτήσεων είναι πιο ανεβασμένο σε σχέση με το περιβάλλον της εφαρμογής».

Εκπαιδευτικός ΠΕ19: Προτάσεις:

- Εμπλουτισμός με περισσότερες θεματικές ενότητες και ερωτήσεις.
- Δυνατότητα δημιουργίας νέων θεματικών ενοτήτων και ερωτήσεων από το χρήστη (π.χ. μέσω μενού διαχειριστή στην εφαρμογή).
- Αξιολόγηση των απαντήσεων που έδωσε ο μαθητής μετά το τέλος χρήσης της εφαρμογής.
- Συγκεντρωτικά αποτελέσματα - σύγκριση με αποτελέσματα άλλων μαθητών, εμφάνιση στατιστικών κλπ.

Εκπαιδευτικός ΠΕ04.01: «Συνίσταται καλύτερη ανάλυση εικόνας και βελτίωση στην κίνηση».

Εκπαιδευτικός ΠΕ20: «Διεπαφή για τους εκπαιδευτικούς, ώστε να μπορούν να βάζουν εύκολα τις δικές τους ερωταπαντήσεις. Αυτό θα επικουρήσει να γίνεται η κατάλληλη προσαρμογή του παιχνιδιού στις απαιτήσεις του εκάστοτε μαθήματος».

Εκπαιδευτικός ΠΕ06: «Τα γραφικά θα μπορούσαν να είναι πιο ελκυστικά».

Εκπαιδευτικός ΠΕ19: -

Εκπαιδευτικός ΠΕ20: «Η εφαρμογή θα μπορούσε να έχει καλύτερα γραφικά που θα κάνουν πιο ελκυστικό το περιβάλλον στους μαθητές. Πολλές φορές βγαίνουν οι ίδιες ερωτήσεις περισσότερες και από 2 φορές».

Εκπαιδευτικός ΠΕ20: «Τα γραφικά της!»

4.4 Ανάλυση αποτελεσμάτων

Κατά την ανάλυση των αποτελεσμάτων, η συντριπτική πλειοψηφία των συμμετεχόντων εκπαιδευτικών είχε θετική στάση από την αρχή, δηλαδή πριν του παρουσιαστεί ο «Λαβύρινθος», ως προς την ένταξη μιας εκπαιδευτικής εφαρμογής σε έξυπνη συσκευή στη διδασκαλία, ακόμα και από εκείνους άλλων ειδικοτήτων. Από τις υπάρχουσες εκπαιδευτικές εφαρμογές υπήρξε μια αρνητική στάση από πλευράς ενός εκπαιδευτικού, ωστόσο, το σύνολο των ερωτηθέντων, συμπεριλαμβανομένου και του προαναφερθέντα, θεωρεί ότι θα ήταν χρήσιμη μία τέτοια εφαρμογή ως συμπλήρωμα στο μάθημα και θα την αξιοποιούσε. Σύμφωνα με το δείγμα, η αξιοποίηση έχει να κάνει κυρίως ως προς την αύξηση του ενδιαφέροντος των μαθητών για το μάθημα, αλλά και σε ένα καλό βαθμό ως προς την αποτελεσματικότερη εμπέδωση των εννοιών. Ένα μικρότερο ποσοστό (45%), πιστεύει ότι η συμπληρωματική χρήση εκπαιδευτικών παιχνιδιών σε κινητές συσκευές θα βοηθούσε και στη βελτίωση της απόδοσης των μαθητών στις εργασίες τους.

Αφού οι συμμετέχοντες εκπαιδευτικοί δοκίμασαν το εκπαιδευτικό παιχνίδι Λαβύρινθος, οι εντυπώσεις τους, όπως προκύπτει από τις αξιολογήσεις τους, φαίνονται πολύ θετικές. Με μέσο

όρο 8,5, ουσιαστικά βρίσκουν την εφαρμογή πολύ ενδιαφέρουσα, ανεξάρτητα από την ειδικότητά τους. Λιγότερο, αλλά αρκετά εύχρηστη βρήκαν τη διεπαφή, με μέσο όρο αξιολόγησης 7,9.

Όλοι οι εκπαιδευτικοί συμφωνούν ότι το εκπαιδευτικό παιχνίδι Λαβύρινθος μπορεί να προσδώσει σημαντικό κίνητρο στους μαθητές για περαιτέρω μελέτη, ενώ κάποιοι πιστεύουν ότι η ενθάρρυνση μπορεί να είναι μεγάλη (27%). Επίσης, πιστεύουν ότι ο «Λαβύρινθος προβάλλει αρκετά (55%) έως πολύ (45%) δυνατότητες που μπορούν παρέχουν οι κινητές πλατφόρμες.

Με μια μέση αποτίμηση των αποτελεσμάτων, οι εκπαιδευτικοί πιστεύουν ότι η εφαρμογή Λαβύρινθος θα τους βοηθούσε αρκετά στον εμπλουτισμό της διδασκαλίας τους, αλλά και τους μαθητές όσον αφορά τη βελτίωση των αποτελεσμάτων τους και την ανάπτυξη των δεξιοτήτων τους στις νέες τεχνολογίες. Πιστεύουν μάλιστα ότι το δικτυακό μέρος τους παιχνιδιού θα μπορούσε να συμβάλλει αρκετά έως πολύ και στην προώθηση της συνεργατικής μάθησης.

Σε γενικές γραμμές, οι εκπαιδευτικοί μετά τη δοκιμή τους, προσδίδουν στη εφαρμογή Λαβύρινθος υψηλή βαθμολογία από ψυχαγωγικής άποψης (μέσος όρος 8,5) και ακόμη υψηλότερη από διδακτικής άποψης (μέσος όρος 8,7). Έτσι, όλοι στο σύνολό τους (100%) δηλώνουν ότι θα χρησιμοποιούσαν τη συγκεκριμένη εφαρμογής στην τάξη τους.

Σίγουρα βέβαια υπάρχουν και κάποιες παρατηρήσεις, προτάσεις για περαιτέρω βελτιώσεις, αλλά και κριτική, τα οποία είναι πάντα απαραίτητα για τη συνεχή πρόοδο, αφού αυτός είναι άλλωστε και ο λόγος της αξιολόγησης από τρίτους. Όλες αυτά θα πρέπει να ληφθούν υπόψη ως συμβουλές, ώστε σε βάθος χρόνου να υπάρξει μια εφαρμογή καλύτερη, καθαρή από σφάλματα και πιο πλούσια σε επιλογές και δυνατότητες.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντικά Σχέδια

Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας τις αρχές της αντικειμενοστρεφούς τεχνολογίας. Αυτό έχει σαν αποτέλεσμα να έχει δημιουργηθεί ένα λογισμικό φορητό, ευέλικτο, συντηρήσιμο και επαναχρησιμοποιήσιμο.

5.1 Συμπεράσματα

Θεωρητικά, για να αναλύσει και σχεδιάσει κάποιος σύμφωνα με τις αρχές της αντικειμενοστρεφούς τεχνολογίας, δεν είναι απαραίτητο να γνωρίζει προγραμματισμό. Στην πράξη όμως, αν γνωρίζει προγραμματισμό, τον βοηθάει να φτιάξει ένα προσχέδιο γρηγορότερα και καλύτερα προσαρμοσμένο στις απαιτήσεις της αναπτυξιακής ομάδας.

Διαπιστώνεται ότι καλύτερη πρακτική είναι να μην ακολουθούνται αυστηρά τα βήματα της μεθοδολογίας, αλλά να γίνεται μεταπήδηση από το ένα στάδιο στο άλλο, όσο συχνά χρειάζεται (εμπρός ή πίσω). Είναι προτιμότερο να διορθώνονται τα λάθη κατά τη διαδικασία ανάπτυξης, κι όχι αφού θα έχει ήδη χτιστεί η εφαρμογή.

5.1.1 Συμπεράσματα της Έρευνας

Από τα προσδοκώμενα αποτελέσματα ικανοποιήθηκαν τα εξής:

- Το λογισμικό είναι αρκετά εύκολο στη χρήση, αλλά θα μπορούσε να ήταν ευκολότερο.
- Αποτιμάται ότι θα μπορέσει να ψυχαγωγήσει και να ταυτόχρονα να μεταφέρει γνώσεις.
- Χρησιμοποιεί απλή ελληνική γλώσσα.
- Οι προσφερόμενες δυνατότητες των κινητών συσκευών (οθόνης αφής, αισθητήρες κλπ.) αξιοποιούνται με διακριτικό τρόπο και συμβάλλουν στη διαδικασία της μάθησης και την ανάπτυξη δεξιοτήτων.
- Ο χρήστης μπορεί εύκολα να διαφύγει από το παιχνίδι οποιαδήποτε στιγμή πατώντας το πλήκτρο «πίσω» (back), που υπάρχει σε όλες τις πλατφόρμες με λειτουργικό σύστημα Android.
- Πιθανότατα θα προωθήσει τη συνεργατική αλλά όχι και την εξατομικευμένη μάθηση.

Γενικά έγινε μια σημαντική προσπάθεια να δημιουργηθεί ένα παιχνίδι με εκπαιδευτικά χαρακτηριστικά στην ελληνική γλώσσα για κινητές συσκευές, με όσο το δυνατόν εύχρηστο τρόπο παρουσίασης της πληροφορίας και προώθηση της συνεργασίας, προκειμένου το παιδί να αποκτήσει κίνητρο προς τη μάθηση. Τα αποτελέσματα δείχνουν ενθαρρυντικά.

5.1.2 Σύνοψη

Βάσει της ανάλυσης των αποτελεσμάτων των ερωτηματολογίων των εκπαιδευτικών θα μπορούσαμε να πούμε ότι η χρήση ενός ψυχαγωγικού συνεργατικού παιχνιδιού με εκπαιδευτικό προσανατολισμό, όπως ο «Λαβύρινθος», μπορεί να βοηθήσει σε σημαντικό βαθμό τόσο τη διαδικασία μάθησης όσο και την ανάπτυξη δεξιοτήτων.

Το συγκεκριμένο παιχνίδι περιέχει ερωτήσεις στην ελληνική γλώσσα για το μάθημα των Νέων Τεχνολογιών, των δύο τελευταίων τάξεων του δημοτικού σχολείου. Επομένως, το παιχνίδι μπορούν να το παίξουν κυρίως οι μαθητές της Πέμπτης και της Έκτης δημοτικού. Απευθύνεται ωστόσο, εκτός από τους αναφερόμενους μαθητές και στους δασκάλους του αντίστοιχου

μαθήματος από την Ελλάδα και την Κύπρο, οι οποίοι μπορούν να το χρησιμοποιήσουν ως συμπληρωματικό βοήθημα για τη διδασκαλία τους.

Το εκπαιδευτικό παιχνίδι Λαβύρινθος έχει ως σκοπό να ψυχαγωγήσει, ταυτόχρονα να εκπαιδεύσει και να βοηθήσει στην εμπέδωση εννοιών. Μπορεί να χρησιμοποιηθεί από το δάσκαλο Πληροφορικής ως συμπλήρωμα για το μάθημά του.

Η ανάλυση, η σχεδίαση και η υλοποίηση της εφαρμογής έγινε σύμφωνα με τις αρχές της αντικειμενοστρεφούς τεχνολογίας. Η σχεδίαση έγινε με τη χρήση των διαγραμμάτων UML, τα οποία σχεδιάστηκαν με τη χρήση του λογισμικού Modelio και η υλοποίηση στη γλώσσα Java χρησιμοποιώντας το βοηθητικό πακέτο SDK (Software Development Kit) του Android. Το περιβάλλον ανάπτυξης (IDE) που χρησιμοποιήθηκε είναι το Android Studio που παρέχει η Google για την ανάπτυξη εφαρμογών για την πλατφόρμα Android. Όλη η διαδικασία της σχεδίασης και υλοποίησης πραγματοποιήθηκε σε Ηλεκτρονικούς Υπολογιστές με λειτουργικό σύστημα Microsoft Windows.

Η εφαρμογή Λαβύρινθος αξιοποιεί αρκετές από τις δυνατότητες που παρέχουν οι σύγχρονες πλατφόρμες κινητού υπολογισμού. Δυνατότητες ασύρματης επικοινωνίας χωρίς την εξάρτηση από άλλες συσκευές με τη χρήση του bluetooth, διεπαφή με τη χρήση των αισθητήρων touchscreen και accelerometer, αλλά και η δυνατότητα δόνησης στην αποτυχία λύσης αινίγματος αξιοποιήθηκαν με διακριτικό κι όχι καταχρηστικό τρόπο. Επιπρόσθετα προβάλλονται εξειδικευμένες χρήσεις των δυνατοτήτων μιας κινητής συσκευής, όπως το προσομοιωμένο ριζίμο του ζαριού με τη χρήση του επιταχυνσιόμετρου (accelerometer).

Η αξιολόγηση του λογισμικού βασίστηκε κατά πλειοψηφία σε αντιλήψεις εκπαιδευτικών με ειδικότητα στην Πληροφορική, πλαισιωμένοι με μια μειοψηφία διαφόρων άλλων ειδικοτήτων για να προσδώσουν μια σφαιρικότερη άποψη. Οι συμμετέχοντες στην αξιολόγηση ήταν εν ενεργεία εκπαιδευτικοί από διαφορετικές περιοχές της Ελλάδας, που διδάσκουν ή έχουν διδάξει σε δημοτικά σχολεία, καθώς η εφαρμογή απευθύνεται στις τελευταίες τάξεις του δημοτικού, με εξαίρεση έναν καθηγητή Φυσικής που έχει διδάξει στις πρώτες τάξεις του Γυμνασίου και έναν Γεωπόνο, καθηγητή που διδάσκει σε Επαγγελματικά Λύκεια (ΕΠΑΛ).

Μετά την αξιολόγηση, η ψυχαγωγική και η διδακτική απόδοση εκτιμούνται αρκετά υψηλές και συγκεκριμένα 8,5 και 8,7 σε κλίμακα 1 - 10 αντίστοιχα.

5.2 Σκέψεις και μελλοντικά σχέδια

Το πρώτο πράγμα που έρχεται στο μυαλό είναι η ολοκλήρωση της υλοποίησης του δικτυακού κομματιού της εφαρμογής. Ήδη λειτουργεί έως ένα βαθμό, όπως έχει περιγραφεί σε προηγούμενο κεφάλαιο, αξιοποιώντας την τεχνολογία bluetooth που διαθέτουν στην συντριπτική τους πλειοψηφία οι πλατφόρμες κινητού υπολογισμού. Η υλοποίηση που πραγματοποιήθηκε δείχνει ότι είναι δυνατή η δημιουργία δικτυακού συνεργατικού παιχνιδιού σε επίπεδο τοπικού δικτύου με τη χρήση του bluetooth. Σε αντίθεση με το wifi, η χρήση του bluetooth καθιστά τη διεπαφή ευκολότερη για το μέσο χρήστη και τη σύνδεση ανεξαρτητοποιημένη από την παρουσία οποιωνδήποτε άλλων ασύρματων δικτύων και πρόσβασης στο διαδίκτυο στην περιοχή. Παρόλα αυτά, η υλοποίηση μιας επιπρόσθετης δυνατότητας σύνδεσης με απομακρυσμένους χρήστες, μέσω του διαδικτύου και τη χρήση του wifi, εξακολουθεί πάντα να είναι μια ευπρόσδεκτη προσθήκη.

Η διεπαφή κατά την προσομοίωση του λαβύρινθου, αν και αξιολογήθηκε θετικά, χρήζει περαιτέρω βελτίωσης. Η εφαρμογή Λαβύρινθος διαθέτει ήδη κάποια στοιχεία κίνησης (animation), ηχητικών εφέ, καθώς και ένα πρωτότυπο μουσικό μοτίβο, προϊόν δικής μας σύνθεσης. Η ανάλυση της οθόνης που χρησιμοποιήθηκε είναι ίδια για όλες τις συσκευές και είναι αρκετά χαμηλή, όπως αναφέρθηκε και στις παρατηρήσεις των αξιολογητών. Το τελευταίο αποφασίστηκε προκειμένου να υποστηρίζονται κάποιες συσκευές με πολύ μικρή οθόνη. Αυτό μπορεί να διορθωθεί, είτε χρησιμοποιώντας εξ αρχής οθόνη και γραφικά στοιχεία υψηλότερης ανάλυσης, είτε να χρησιμοποιείται πλήθος από διαφορετικές αρχικές αναλύσεις και στοιχεία, αντίστοιχα με τη γηγενή ανάλυση που υποστηρίζεται από τη συσκευή. Σε αυτά θα μπορούσε να προστεθεί και μια αναβάθμιση που έχει σχέση με τη βελτίωση της ομαλότητας του animation και της σύνθεσης του μουσικού μοτίβου, ώστε να γίνουν ακόμα πιο ελκυστικά για τα παιδιά.

Σημαντικό θέμα που αναφέρθηκε στην αξιολόγηση και πρέπει να ληφθεί υπόψη είναι η όσο το δυνατόν εξάλειψη των σφαλμάτων (bugs), καθώς κάποιες φορές, φαινομενικά τυχαίες, μπορεί το λογισμικό να βγάλει σφάλμα και να κλείνει απότομα ή να εμφανίζει τον χαρακτήρα του παιχνιδιού σε αδιέξοδο. Μπορεί η επανεκκίνηση του προγράμματος να λύνει προσωρινά το πρόβλημα, αλλά θα πρέπει να μπει κι αυτό στη λίστα των μελλοντικών βελτιώσεων.

Επίσης, στην εφαρμογή δεν υποστηρίζεται αποθήκευση του high-score. Μια μελλοντική προσθήκη αυτής της δυνατότητας, μαζί με την εμφάνιση συγκεντρωτικών αποτελεσμάτων και συγκριτικών με άλλους παίκτες, θα προσέδιδε περισσότερο κίνητρο στα παιδιά, καθώς τους

αρέσει ο ανταγωνισμός, όπως ανέφερε και ένας από τους αξιολογητές εκπαιδευτικούς. Μία ακόμη προτεινόμενη βελτίωση που έχει σχέση με τα παραπάνω, είναι η αξιολόγηση των απαντήσεων που έδωσε ο μαθητής μετά το τέλος χρήσης του παιχνιδιού.

Όσον αφορά τις ερωτήσεις θα μπορούσαν να προσαρμοστούν λίγο καλύτερα στο επίπεδο των μαθητών. Ακόμα καλύτερα θα μπορούσε να προστεθεί μια διεπαφή με δυνατότητα διαχείρισης των θεματικών ενοτήτων και των ερωτήσεων από το χρήστη εκπαιδευτικό, ώστε να μπορεί να προσθέτει, να τροποποιεί και να διαγράφει ό,τι δεδομένο από τη βάση εκείνος επιθυμεί. Αυτό θα μπορούσε να υλοποιηθεί με την υλοποίηση ενός μενού διαχείρισης. Αυτό θα επικουρήσει όχι μόνο στο να γίνεται η κατάλληλη προσαρμογή του παιχνιδιού στις απαιτήσεις του εκάστοτε μαθήματος, αλλά θα επεκτείνει τη χρήση του παιχνιδιού ως βοήθημα και σε άλλα μαθήματα.

Η έρευνα και ανάπτυξη γενικότερα έχει χώρο για συνέχιση και βελτίωση. Η αξιολόγηση μπορεί να επεκταθεί και να γίνει από ακόμη περισσότερους εκπαιδευτικούς, αλλά και από μαθητές. Ασφαλέστερα αποτελέσματα για την έρευνα, οπωσδήποτε θα προκύψουν εφαρμόζοντάς τη σε ένα πλήθος από πραγματικές σχολικές τάξεις.

Τέλος, μία ακόμη σκέψη είναι γίνει μετάφραση της διεπαφής στην αγγλική γλώσσα, ώστε η εφαρμογή να δοκιμαστεί από περισσότερους χρήστες, αλλά και να αξιολογηθεί περαιτέρω από διεθνή αποθετήρια.

5.3 Επίλογος

Οι έξυπνες κινητές συσκευές με τις οποίες είναι πλέον εξοικειωμένη η πλειοψηφία των νέων, έχουν τη δυναμική να διευκολύνουν τη διαδικασία της μάθησης, εφόσον αξιοποιηθούν σωστά. Ο «Λαβύρινθος» παρέχει ένα ελκυστικό περιβάλλον που δίνει κίνητρο στα παιδιά για μάθηση και σημαντικός παράγοντας για την επίτευξη του τελευταίου είναι η χρήση μαθησιακών δραστηριοτήτων που να αξιοποιούν τις ιδιαίτερες δυνατότητες των συσκευών αυτών.

Η ανάπτυξη της συγκεκριμένης εφαρμογής, εκτός από την πιθανή χρήση της ως βοήθημα στη διδασκαλία, συνέβαλε και στην απόκτηση τεχνογνωσίας, η οποία μπορεί να αξιοποιηθεί για τις μελλοντικές βελτιώσεις της υπάρχουσας εφαρμογής, αλλά και για την ευκολότερη και ταχύτερη ανάπτυξη νέων στις πλατφόρμες κινητού υπολογισμού με λειτουργικό σύστημα Android.

Παράρτημα Α

Ερωτηματολόγια Εκπαιδευτικών

A.1 Ερωτηματολόγιο εκπαιδευτικών πριν την υλοποίηση

1. Ειδικότητα εκπαιδευτικού:

2. Πιστεύετε πως θα ήταν χρήσιμο να υπάρχει μια εφαρμογή σε έξυπνη συσκευή (smartphone, smart device) ως συμπλήρωμα στο μάθημα;

- a. Ναι
- b. Όχι
- c. Ίσως

3. Έχετε εντάξει ή σκέφτεστε να εντάξετε υπάρχουσες εκπαιδευτικές εφαρμογές στο μάθημά σας;
- a. Ναι
 - b. Όχι
4. Θα αξιοποιούσατε μία τέτοια εφαρμογή;
- a. Ναι
 - b. Όχι
5. Ποιο θεωρείτε ότι μπορεί να είναι το αποτέλεσμα της συμπληρωματικής χρήσης εκπαιδευτικών παιχνιδιών σε κινητές συσκευές στο μάθημά σας; (Μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις)
- a. Θα αυξήσει το ενδιαφέρον των μαθητών για το μάθημα σας
 - b. Θα βελτιώσει την απόδοση των μαθητών στις εργασίες τους
 - c. Θα βοηθήσει στην αποτελεσματικότερη εμπέδωση των εννοιών

A.2 Ερωτηματολόγιο εκπαιδευτικών μετά την υλοποίηση

Στη συνέχεια παρατίθενται οι απόψεις 10 εκπαιδευτικών, όπως καταγράφηκαν σε ερωτηματολόγια για την αποτύπωση των εντυπώσεών τους από τη χρήση του Λαβύρινθου. Οι ερωτήσεις εστίασαν στην άποψή τους για το Λαβύρινθο ως ψυχαγωγικό εργαλείο για τη διδασκαλία και τη μάθηση εννοιών πληροφορικής στις τελευταίες τάξεις του δημοτικού, τη δεκτικότητά τους όσον αφορά την αξιοποίηση των τεχνολογιών στις πλατφόρμες κινητού υπολογισμού, στον εντοπισμό του κάποιου σημείου που τους προβληματίζει και στις προτάσεις τους για τυχόν βελτιώσεις.

1. Πόσο ενδιαφέρουσα βρήκατε την εφαρμογή σε κλίμακα 1 – 10;

2. Πόσο εύχρηστη βρήκατε τη διεπαφή της εφαρμογής σε κλίμακα 1 – 10;

3. Πιστεύετε ότι ο «Λαβύρινθος» μπορεί να ενθαρρύνει τους μαθητές για περαιτέρω μελέτη;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ

4. Πιστεύετε ότι ο «Λαβύρινθος» προβάλλει δυνατότητες που μπορούν να παρέχουν οι κινητές πλατφόρμες;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ
- e. Υπερβολικά

5. Πιστεύετε ότι θα σας βοηθούσε η εφαρμογή στον εμπλουτισμό της διδασκαλίας σας;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ

6. Πιστεύετε ότι θα σας βοηθούσε η εφαρμογή στη βελτίωση των αποτελεσμάτων των μαθητών σας;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ

7. Πιστεύετε ότι η χρήση ενός τέτοιου ψυχαγωγικού παιχνιδιού σε κινητές πλατφόρμες μπορεί να βοηθήσει στην ανάπτυξη δεξιοτήτων στις νέες τεχνολογίες;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ

8. Πιστεύετε ότι η χρήση του δικτυακού μέρους ενός τέτοιου ψυχαγωγικού συνεργατικού παιχνιδιού μπορεί να προωθήσει τη συνεργατική μάθηση;

- a. Καθόλου
- b. Λίγο
- c. Αρκετά
- d. Πολύ

9. Πόσο θα βαθμολογούσατε την εφαρμογή από διδακτικής άποψης σε κλίμακα 1 – 10;

10. Πόσο θα βαθμολογούσατε την εφαρμογή από ψυχαγωγικής άποψης σε κλίμακα 1 – 10;

- 11.** Θα χρησιμοποιούσατε την εφαρμογή στην τάξη σας;
- a. Ναι
 - b. Όχι
- 12.** Αν απαντήσατε «Όχι» στην προηγούμενη ερώτηση, παρακαλώ αιτιολογήστε.
- 13.** Υπάρχει κάποια περαιτέρω βελτίωση στην εφαρμογή που θα θέλατε να προτείνετε;

Παράρτημα Β

Αρχεία XML

Παρατίθεται το περιεχόμενο των xml αρχείων που χρησιμοποιήθηκαν για την εφαρμογή *Λαβύρινθος*.

B.1 Το αρχείο manifest

Το απαραίτητο αρχείο `AndroidManifest.xml` περιγράφει βασικές πληροφορίες για τη δομή και τις απαιτήσεις της εφαρμογής στο Android.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="grpekast.sch.users.labyrinthos" >

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-feature android:name="android.hardware.screen.landscape" />.

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
```

```

        android:name=".Main"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:configChanges="orientation|keyboardHidden"
        android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>

    <activity
        android:name=".Platform"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:configChanges="orientation|keyboardHidden"
        android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" >
    </activity>

    <activity
        android:name="grpekast.sch.users.network.Syndesi"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:configChanges="orientation|keyboardHidden"
        android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" >
    </activity>
</application>
</manifest>

```

B.2 Activities

Στο παραπάνω αρχείο manifest αναφέρονται τα τρία activities που υλοποιούνται στην εφαρμογή, δηλαδή τα Main, Platform και Syndesi. Τα γραφικά στοιχεία απεικόνισής τους περιγράφονται από τα αρχεία xml *activity_labyrinthos.xml*, *activity_game.xml*, *activity_syndesi.xml*, αντίστοιχα:

activity_labyrinthos.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/welcome"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="@drawable/welcome">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center|left"
        android:weightSum="1"
        android:baselineAligned="false">
        <LinearLayout
            android:orientation="vertical"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:layout_gravity="center_vertical"

```

```

        android:layout_marginLeft="16dp">
    <ImageView
        android:id="@+id/startButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/start_button"
        android:layout_marginTop="80dp"
        android:contentDescription="@string/button"/>
    <ImageView
        android:id="@+id/joinButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/join_button"
        android:layout_marginTop="40dp"
        android:contentDescription="@string/button"/>
    </LinearLayout>
</LinearLayout>
</FrameLayout>

```

activity_game.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/game_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <VideoView
        android:id="@+id/video_player_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentRight="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"/>
</RelativeLayout>

```

activity_syndesi.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="#40000000">
    <TableLayout
        android:id="@+id/syndesi"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="top|center"
        android:stretchColumns="1"
        android:paddingLeft="24dip"
        android:paddingTop="12dip"
        android:paddingRight="24dip"
        android:paddingBottom="12dip">
        <TableRow>
            <TextView
                android:id="@+id/syndesi_title"
                android:layout_column="0"
                android:layout_span="2"
                android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="@string/join_title"
        android:textSize="@dimen/text_size_title"
        android:padding="4dip"
        android:textStyle="bold"/>

        <Button
            style="?android:attr/buttonStyleSmall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/ΕΥΝΕΧΕΙΑ..."
            android:id="@+id/serverButton"
            android:layout_column="2" />

    </TableRow>
    <TableRow>
        <View
            android:layout_width="match_parent"
            android:layout_height="3dip"
            android:layout_span="3"
            android:background="#FF909090"/>
    </TableRow>
</TableLayout>
</ScrollView>

```

B.3 Resource References

Σε αυτά τα αρχεία xml, δηλώνονται αναφορές για προκαθορισμένους πόρους, όπως το θέμα εμφάνισης (*styles.xml*) και τιμές σταθερών συμβολοσειρών (*strings.xml*), προς χρήση από την εφαρμογή.

styles.xml

```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
    </style>

    <style name="Theme.AppCompat.Light.NoActionBar.FullScreen"
        parent="Theme.AppCompat.Light.NoActionBar">
        <item name="android:windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="android:windowFullscreen">true</item>
        <item name="android:windowContentOverlay">@null</item>
    </style>

</resources>

```


strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Λαβύρινθος</string>

  <string name="action_settings">Settings</string>
  <string name="button">button</string>
  <string name="join_title">"ΔΙΑΘΕΣΙΜΟΙ ΠΑΙΚΤΕΣ"</string>
</resources>
```

B.4 Προσαρμοσμένα Κουμπιά

Προσομοιώνονται τρία γραφικά κουμπιά για τη διεπαφή. Αυτά είναι τα *start_button.xml* (ΑΤΟΜΙΚΟ ΠΑΙΧΝΙΔΙ), *join_button.xml* (ΟΜΑΔΙΚΟ ΠΑΙΧΝΙΔΙ) και *syndesi_button.xml* (ΣΥΝΔΕΣΗ).

start_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/singleplayer_pressed"
android:state_pressed="true"/>
  <item android:drawable="@drawable/singleplayer"/>
</selector>
```

join_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/multiplayer_pressed"
android:state_pressed="true"/>
  <item android:drawable="@drawable/multiplayer"/>
</selector>
```

syndesi_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:state_pressed="true"
    android:drawable="@drawable/btn_syndesi_pressed"/>
  <item
    android:state_pressed="false"
    android:drawable="@drawable/btn_syndesi"/>
</selector>
```

Παράρτημα Γ

Κώδικας Java

Στη συνέχεια παρατίθεται ο πλήρης κώδικας Java.

Γ.1 Αρχείο Main.java

```
/**
 * *****
 * // Java Activity Class Name : Main
 * *****
 */
package grpekast.sch.users.labyrinthos;

import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.Toast;

public class Main extends ActionBarActivity {
    public static boolean multiplayer = false;
    private static final String TAG = "Labyrinthos";
    private ImageView mSinglePlayer, mMultiPlayer;
    private BluetoothAdapter mBluetoothAdapter = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
Log.d(TAG, "onCreate(Bundle) called");

setContentView(R.layout.activity_labyrinthos);
getWindow().addFlags((WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON));

mSinglePlayer = (ImageView) findViewById(R.id.startButton);
mSinglePlayer.setOnClickListener((new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        multiplayer = false;
        Intent myIntent = new Intent(Main.this, Platform.class);
        startActivity(myIntent);
    }
}));
mMultiPlayer = (ImageView) findViewById(R.id.joinButton);
mMultiPlayer.setOnClickListener((new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        multiplayer = true;
        onBluetooth();
        Intent myIntent = new Intent(Main.this, Platform.class);
        startActivity(myIntent);
    }
}));
}

@Override
public void onStart() {
    super.onStart();
    Log.d(TAG, "onStart() called");
}
@Override
public void onPause() {
    super.onPause();
    Log.d(TAG, "onPause() called");
}
@Override
public void onResume() {
    super.onResume();
    Log.d(TAG, "onResume() called");
}
@Override
public void onStop() {
    super.onStop();
    Log.d(TAG, "onStop() called");
}
@Override
public void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "onDestroy() called");
    multiplayer = false;
    offBluetooth();
}

private void offBluetooth() {
    if (mBluetoothAdapter == null) return;
    if (mBluetoothAdapter.isEnabled()) {
        mBluetoothAdapter.disable();
        Log.i("Log", "Bluetooth is Disabled");
    }
}

private void onBluetooth() {
    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if (mBluetoothAdapter == null) {
        // Device does not support Bluetooth
        Toast.makeText(this, "Αυτή η συσκευή δεν υποστηρίζει Bluetooth"
            , Toast.LENGTH_LONG).show();
    }
    return;
}

```

```

    }
    if (!mBluetoothAdapter.isEnabled()) {
        mBluetoothAdapter.enable();
        long t1 = System.nanoTime(), t2;
        while (!mBluetoothAdapter.isEnabled()) {
            t2 = System.nanoTime();
            if (t2 - t1 > 100000000000L) break; }
    }
    Log.i("Log", "Bluetooth is Enabled");
}

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

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Γ.2 Αρχείο Platform.java

```

//*****
// Java Class Name : Platform
//*****
package grpekast.sch.users.labyrinthos;

import android.annotation.TargetApi;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.content.res.AssetManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.Vibrator;
import android.provider.Settings;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.Surface;
import android.view.WindowManager;
import android.widget.FrameLayout;
import android.widget.RelativeLayout;
import android.widget.VideoView;

import java.util.LinkedList;
import java.util.List;

import grpekast.sch.users.network.Bluetooth;
import grpekast.sch.users.network.Diktyo;
import grpekast.sch.users.network.Syndesi;

```

```

//Game Main Activity
public class Platform extends ActionBarActivity {
    public static final int GAME_WIDTH = 320; // 320x240
    public static final int GAME_HEIGHT = 240; // Game
    public static Game sGame; // SurfaceView
    public static AssetManager assets;
    public static Vibrator vibrator;
    public static int iRotation = Surface.ROTATION_90;

    // Message types sent from the Bluetooth Handler
    public static final int MESSAGE_STATE_CHANGE = 1;
    public static final int MESSAGE_READ = 2;
    public static final int MESSAGE_WRITE = 3;
    public static final int MESSAGE_DEVICE_NAME = 4;
    public static final int MESSAGE_TOAST = 5;
    // Key names received from the Bluetooth Handler
    public static final String DEVICE_NAME = "device_name";
    public static final String TOAST = "toast";
    // Connection Handler
    private MessageHandler mHandler = new MessageHandler();
    public static List<Object> sObject = new LinkedList<>();
    private Bluetooth mBluetooth = null;
    private BluetoothAdapter mBluetoothAdapter = null;

    //-----
    // Activity Launcher
    //-----
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        rythmizei();
    }
    @Override
    public void onStart() {
        super.onStart();
        Log.d("Platform", "onStart() called");

        if (Main.multiplayer && mBluetooth == null) {
            // Initialize the Bluetooth to perform bluetooth connections
            mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

            Intent i = new Intent(this, Syndesi.class);
            startActivityForResult(i, 1);

            mBluetooth = new Bluetooth(this, mHandler);
        }
    }
    @Override
    public void onPause() {
        super.onPause();
        Log.d("Platform", "onPause() called");
    }
    @Override
    public synchronized void onResume() {
        super.onResume();
        Log.d("Platform", "onResume() called");
        if (mBluetooth != null) {
            // Only if the state is STATE_NONE, do we know that we haven't started already
            if (mBluetooth.getState() == Bluetooth.STATE_NONE) {
                // Start the Bluetooth services
                mBluetooth.start();
            }
        }
    }
    @Override
    public void onStop() {
        super.onStop();
        Log.d("Platform", "onStop() called");
    }
}

```

```

@Override
public void onDestroy() {
    super.onDestroy();
    Log.d("Platform", "onDestroy() called");
    sObject.clear();
    // Stop the Bluetooth services
    if (mBluetooth != null) mBluetooth.stop();
}

private void rythmizei() {
    vibrator = (Vibrator) this.getSystemService(VIBRATOR_SERVICE);
    assets = getAssets();
    sObject.clear();

    setContentView(R.layout.activity_game);

    RelativeLayout gameView = (RelativeLayout) findViewById(R.id.game_view);
    VideoView videoView = (VideoView) findViewById(R.id.video_player_view);
    sGame = new Game(this, GAME_WIDTH, GAME_HEIGHT, videoView);
    gameView.addView(sGame);

    getWindow().addFlags((WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON));
    getRotation();
}

private void getRotation() {
    String userRotation = Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB ?
        getUserRotationString() : "user_rotation";
    try { iRotation =
        Settings.System.getInt(this.getContentResolver(), userRotation);
    } catch (Settings.SettingNotFoundException e) { e.printStackTrace(); }
    Log.d("Platform", "iRotation = " + iRotation);
}

@TargetApi(Build.VERSION_CODES.HONEYCOMB)
protected String getUserRotationString() {return Settings.System.USER_ROTATION;}

//-----
// Static Inner Class Name : MessageHandler
//-----
static class MessageHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case MESSAGE_STATE_CHANGE:
                Log.i("msg.what", "MESSAGE_STATE_CHANGE: " + msg.arg1);
                switch (msg.arg1) {
                    case Bluetooth.STATE_CONNECTED:
                        Log.i("msg.arg1", "STATE_CONNECTED");
                        break;
                    case Bluetooth.STATE_CONNECTING:
                        Log.i("msg.arg1", "STATE_CONNECTING");
                        break;
                    case Bluetooth.STATE_LISTEN:
                        Log.i("msg.arg1", "STATE_LISTEN");
                    case Bluetooth.STATE_NONE:
                        Log.i("msg.arg1", "STATE_NONE");
                        break;
                }
                break;
            case MESSAGE_WRITE:
                Log.i("msg.what", "MESSAGE_WRITE");
                break;
            case MESSAGE_READ:
                Log.i("msg.what", "MESSAGE_READ");
                sObject.add(msg.obj);
                break;
            case MESSAGE_DEVICE_NAME:
                Log.i("msg.what", msg.getData().getString(DEVICE_NAME));
                break;
            case MESSAGE_TOAST:

```

```

        Log.i("msg.what", "MESSAGE_TOAST");
        break;
    default:
    }
}

//-----
// onActivityResult
//-----
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    Log.d("Platform", "onActivityResult " + resultCode);
    if (requestCode == 1) {
        // When DeviceListActivity returns with a device to connect
        switch (resultCode) {
            case RESULT_OK:
                // Get the device MAC address
                String address = data.getExtras()
                    .getString(Syndesi.REMOTE_DEVICE_ADDRESS);
                // Get the BluetoothDevice object
                BluetoothDevice device = BluetoothAdapter.getDefaultAdapter()
                    .getRemoteDevice(address);
                // Attempt to connect to the device
                mBluetooth.connect(device);
                break;
            case RESULT_CANCELED:
            }
        }
        sGame.setBluetooth(mBluetooth);
    }
}

public Diktyo diktyo;
public void shows() {
    //setWallpaper();
}
public void connect() { }
public void entopizei() { }
}

```

Γ.3 Αρχείο Game.java

```

//*****
// Java Class Name : Game
//*****
package grpekast.sch.users.labyrinthos;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Rect;
import android.media.MediaPlayer;
import android.net.Uri;
import android.util.AttributeSet;
import android.util.Log;
import android.view.MotionEvent;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.content.Context;
import android.view.View;
import android.widget.VideoView;

import grpekast.sch.users.database.Basi;
import grpekast.sch.users.database.Enigma;
import grpekast.sch.users.model.Score;
import grpekast.sch.users.network.Bluetooth;

```

```

import grpekast.sch.users.state.Avatar;
import grpekast.sch.users.state.State;
import grpekast.sch.users.utility.Accelerometer;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.TouchScreen;

public class Game extends SurfaceView implements Runnable {
    private Bitmap gameImage;
    private Rect gameImageSrc;
    private Rect gameImageDst;
    private Canvas gameCanvas;
    private Grafika grafika;

    private Thread gameThread;
    private volatile boolean running = false;
    public volatile boolean paused = true;
    private final Object pauseLock = new Object();
    private volatile State currentState;

    private Basi basi;
    private VideoView video;

    private TouchScreen inputHandler;
    private Accelerometer accelerometerHandler;
    private Bluetooth mBluetooth = null;

    private String packageName;

    //-----
    // Constructors
    //-----
    public Game(Context context, AttributeSet attributeSet) {
        super(context, attributeSet); }
    public Game(final Context context, int gameWidth, int gameHeight, VideoView vView) {
        super(context);
        gameImage = Bitmap.createBitmap(gameWidth, gameHeight, Bitmap.Config.RGB_565);
        gameImageSrc = new Rect(0, 0, gameImage.getWidth(), gameImage.getHeight());
        gameImageDst = new Rect();
        gameCanvas = new Canvas(gameImage);
        grafika = new Grafika(gameCanvas);
        video = vView;
        packageName = context.getPackageName(); //grpekast.sch.users.labyrinthos

    //-----
    // SurfaceHolder.Callback() (Anonymous function)
    //-----
    SurfaceHolder holder = getHolder();
    holder.addCallback(new SurfaceHolder.Callback() {
        @Override
        public void surfaceCreated(SurfaceHolder holder) {
            if (!Game.this.isEnabled()) return;
            basi = new Basi(context);
            initInput(context);
            initVideo();
            setWillNotDraw(false);
            if (currentState == null) setCurrentState(new Avatar(mBluetooth));
            startThread();
            Log.d("Game", "Surface created");
        }

        @Override
        public void surfaceChanged(SurfaceHolder holder, int format
            , int width, int height) {
            Log.d("Game", "Surface changed");
        }

        @Override
        public void surfaceDestroyed(SurfaceHolder holder) {
            if (!Game.this.isEnabled()) return;

```



```

        Log.d("Game", "Surface destroyed");
        stopThread();
        basi.close();
        accelerometerHandler.stopSensor();
        Score.reset();
        System.gc();
    }
});
}

//-----
// run
//-----
/**Starts executing the active part of the class' code. This method is
 * called when a thread is started that has been created with a class which
 * implements {@code Runnable}.*
@Override
public void run() {
    Log.d("Thread", "Created...");
    //These variables should sum up to 25 ms on every iteration (~40FPS)
    long updateDuration = 0; // Measures both update AND render (nsec)
    long sleepDuration = 0; //Measures sleep
    while (running) {
        while (paused) {
            synchronized (pauseLock) {
                try { pauseLock.wait(); }
                catch (InterruptedException e) {}
            }
        }
        long beforeUpdateRender = System.nanoTime();
        long dt = sleepDuration + updateDuration;
        show(dt);
        updateDuration = (System.nanoTime() - beforeUpdateRender) / 1000000L;
        sleepDuration = Math.max(2, 25 - updateDuration);
        try {
            Thread.sleep(sleepDuration);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

//-----
// onDraw
@Override //-----
protected void onDraw(Canvas canvas) {
    show(0);
}

//-----
// show
//-----
private void show(long dt) {
    currentState.update(dt / 1000f);
    currentState.render(grafika);
    show();
}
private void show() {
    Canvas screen;
    screen = getHolder().lockCanvas();
    if (screen != null) {
        screen.getClipBounds(gameImageDst);
        synchronized (getHolder()) {
            screen.drawBitmap(gameImage, gameImageSrc, gameImageDst, null);
            getHolder().unlockCanvasAndPost(screen); }
    }
}

//-----
// setCurrentState

```

```

//-----
public void setCurrentState(State newState) {
    System.gc(); this.setEnabled(true);
    currentState = newState;
    if (playVideo()) return;
    inputHandler.setCurrentState(currentState);
    accelerometerHandler.setCurrentState(currentState);
    currentState.setBackground(grafika);
    if (currentState.isThreaded) { resumeThread(); }
    else { if (paused) invalidate();
           else pauseThread(); }
}

//-----
// setBluetooth
//-----
public void setBluetooth(Bluetooth bt) { mBluetooth = bt; }

//-----
// pauseThread
//-----
public synchronized void pauseThread() {
    synchronized (pauseLock) {
        paused = true;
    }
}

//-----
// resumeThread
//-----
public synchronized void resumeThread() {
    if (!paused) return;
    synchronized (pauseLock) {
        paused = false;
        pauseLock.notifyAll();
    }
}

//-----
// initInput
//-----
private void initInput(Context context) {
    if (inputHandler == null) { inputHandler = new TouchScreen(); }
    setOnTouchListener(inputHandler);
    if (accelerometerHandler == null) {
        accelerometerHandler = new Accelerometer(context); }
}

//-----
// initVideo
//-----
private void initVideo() {
    video.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            stopVideo();
            return false;
        }
    });
    video.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
        @Override
        public void onCompletion(MediaPlayer mp) {
            stopVideo();
        }
    });
}

//-----
// playVideo
//-----

```

```

private boolean playVideo() {
    String strIntroFile = currentState.fetchIntroFile().split("\\.")[0];

    if (strIntroFile.isEmpty())
        return false;
    else { grafika.clear(Color.BLACK); show();
        final String path = "android.resource://" + packageName + "/"
            + getResources().getIdentifier(strIntroFile, "raw", packageName);
        Platform myActivity = (Platform)getContext();
        myActivity.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                Game.this.setEnabled(false);
                Game.this.setVisibility(INVISIBLE);
                video.setVisibility(VISIBLE);
                video.setVideoURI(Uri.parse(path));
                video.start(); }
        });
        return true;
    }
}

//-----
// stopVideo
//-----
private void stopVideo() {
    grafika.clear(Color.BLACK); show();
    video.stopPlayback();
    video.setVisibility(GONE);
    setVisibility(VISIBLE);
    setEnabled(true);
    setCurrentState(currentState);
}

//-----
// startThread
//-----
private void startThread() {
    running = true;
    gameThread = new Thread(this);
    gameThread.start();
}

//-----
// stopThread
//-----
private void stopThread() {
    if (paused) resumeThread();
    running = false;
    while (gameThread.isAlive()) {
        try {
            gameThread.join();
            break; }
        catch (InterruptedException e) {}
    }
}

//-----
// Database connection interface for friend class Enigma
//-----
public class DBConnection { // inner class DBConnection
    private DBConnection() {} // no public constructor
    public Basi getDBHelper() { return basi; }
}
public void openDBConnection(Enigma enigma) {
    enigma.receiveBasi(new DBConnection());
}
}

```

Γ.4 Αρχείο Assets.java

```

//*****
// Java Class Name : Assets
//*****
package grpekast.sch.users.labyrinthos;

import android.annotation.TargetApi;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Typeface;
import android.media.AudioAttributes;
import android.media.AudioManager;
import android.media.SoundPool;
import android.os.Build;

import java.io.IOException;
import java.io.InputStream;

import grpekast.sch.users.utility.Emfanisi;

public class Assets {
    private static SoundPool soundPool;
    public static Bitmap score, scoreDown, box, boxOpen;
    public static Bitmap grass, cloud, sphinxMedium, heart, diceIsometric;
    public static Bitmap[] beastie = new Bitmap[24], sphinx = new Bitmap[5];
    public static Bitmap[] exit = new Bitmap[4], dice = new Bitmap[6];
    public static int labyrinthosID, startID, gameOverID;
    public static int sfiggaID, correctID, wrongID, successID, lifeID, latchID;
    public static int bonusFasiID, boxOpenID, diceID, lifeEmptyID;
    public static Typeface comicsans;
    private static boolean IsLoaded = false;

    //-----
    // load (assets)
    //-----
    public static void load() {
        if (IsLoaded) return;

        // Various images
        cloud = loadBitmap("cloud.png", true);
        heart = loadBitmap("kardoula.png", true);
        box = loadBitmap("kouti.png", true);
        boxOpen = loadBitmap("kouti_open.png", true);

        // Avatars
        beastie[0] = loadBitmap("ageladitsa.png", true);
        beastie[1] = loadBitmap("alepouditsa.png", true);
        beastie[2] = loadBitmap("arkoudaki.png", true);
        beastie[3] = loadBitmap("vatraxi.png", true);
        beastie[4] = loadBitmap("gatoulal.png", true);
        beastie[5] = loadBitmap("gatoula2.png", true);
        beastie[6] = loadBitmap("gatoulis.png", true);
        beastie[7] = loadBitmap("gourounaki.png", true);
        beastie[8] = loadBitmap("ippokampos.png", true);
        beastie[9] = loadBitmap("kotoula.png", true);
        beastie[10] = loadBitmap("kounelaki1.png", true);
        beastie[11] = loadBitmap("kounelaki2.png", true);
        beastie[12] = loadBitmap("maimoudaki.png", true);
        beastie[13] = loadBitmap("panda.png", true);
        beastie[14] = loadBitmap("papaki.png", true);
        beastie[15] = loadBitmap("penguinaki1.png", true);
        beastie[16] = loadBitmap("penguinaki2.png", true);
        beastie[17] = loadBitmap("petalouditsa.png", true);
        beastie[18] = loadBitmap("pontikaki.png", true);
        beastie[19] = loadBitmap("poulaki.png", true);
        beastie[20] = loadBitmap("probataki.png", true);
    }
}

```

```

beastie[21] = loadBitmap("skylaki1.png", true);
beastie[22] = loadBitmap("skylaki2.png", true);
beastie[23] = loadBitmap("psaraki.png", true);

// Sphinxes
sphinx[0] = loadBitmap("sfigga.png", true);
sphinx[1] = Emfanisi.color1(sphinx[0]);
sphinx[2] = Emfanisi.color2(sphinx[0]);
sphinx[3] = Emfanisi.color3(sphinx[0]);
sphinx[4] = Emfanisi.color4(sphinx[0]);
sphinxMedium = loadBitmap("sfigga_medium.png", true);

// Exits (chests, doors)
exit[0] = loadBitmap("sentouki.png", true);
exit[1] = loadBitmap("sentouki_open.png", true);
exit[2] = loadBitmap("porta.png", true);
exit[3] = loadBitmap("porta_open.png", true);

// Dice
for (int i = 0; i < 6; i++)
    dice[i] = loadBitmap("zari" + (i+1) + ".png", false);
diceIsometric = loadBitmap("zari.png", true);

// Sounds
sfiggaID = loadSound("sfigga.ogg");
correctID = loadSound("select.ogg");
wrongID = loadSound("wrong.ogg");
successID = loadSound("success.ogg");
lifeID = loadSound("heart.ogg");
lifeEmptyID = loadSound("life_empty.ogg");
latchID = loadSound("latch.ogg");
boxOpenID = loadSound("box_open.ogg");
labyrinthosID = loadSound("labyrinthos.ogg");
startID = loadSound("start.ogg");
bonusFasiID = loadSound("appear.ogg");
diceID = loadSound("dice_shake.ogg");
gameOverID = loadSound("game_over.ogg");

// Font
comicsans = loadFont("comicbd.ttf");

IsLoaded = true;
}

//-----
// loadBitmap
//-----
private static Bitmap loadBitmap(String filename, boolean transparency) {
    InputStream inputStream = null;

    try { inputStream = Platform.assets.open(filename); }
    catch (IOException e) { e.printStackTrace(); }

    BitmapFactory.Options options = new BitmapFactory.Options();

    if (transparency) {
        options.inPreferredConfig = Bitmap.Config.ARGB_8888; }
    else {
        options.inPreferredConfig = Bitmap.Config.RGB_565;
    }
    Bitmap bitmap = BitmapFactory.decodeStream(inputStream, null, options);
    return bitmap.copy(options.inPreferredConfig, true);
}

private static int loadSound(String filename) {
    int soundID = 0;
    if (soundPool == null) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            createNewSoundPool(); }
        else { createOldSoundPool(); }
    }
}

```

```

    }
    try {
        soundID = soundPool.load(Platform.assets.openFd(filename), 1); }
    catch (IOException e) {
        e.printStackTrace();
    }
    return soundID;
}

//-----
// loadFont
//-----
private static Typeface loadFont(String filename) {
    return Typeface.createFromAsset(Platform.assets, filename);
}

//-----
// playSound
//-----
public static void playSound(int soundID) {
    soundPool.play(soundID, 1, 1, 1, 0, 1);
}

//-----
// createSoundPool
//-----
@TargetApi(Build.VERSION_CODES.LOLLIPOP)
protected static void createNewSoundPool() {
    AudioAttributes attributes = new AudioAttributes.Builder()
        .setUsage(AudioAttributes.USAGE_GAME)
        .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
        .build();
    soundPool = new SoundPool.Builder()
        .setAudioAttributes(attributes)
        .build();
}
@SuppressWarnings("deprecation")
protected static void createOldSoundPool() {
    soundPool = new SoundPool(25, AudioManager.STREAM_MUSIC, 0);
}
}

```

Γ.5 Αρχείο SpatialHashGrid.java

```

//*****
// Java Class Name : SpatialHashGrid
//*****
package grpekast.sch.users.labyrinthos;

import java.util.ArrayList;

import grpekast.sch.users.model.Stoixeio;

public class SpatialHashGrid {
    ArrayList<Stoixeio>[] dynamicCells;
    ArrayList<Stoixeio>[] staticCells;
    int cellsPerRow;
    int cellsPerCol;
    float cellSize;
    float negativeY;
    int[] cellIds = new int[4];
    ArrayList<Stoixeio> foundObjects;

    //-----
    // Constructor

```

```

//-----
public SpatialHashGrid(float worldTop, float worldWidth, float worldHeight
, float cellSize) {
    this.negativeY = worldTop >= 0 ? 0 : -worldTop;
    this.cellSize = cellSize;
    this.cellsPerRow = (int) Math.ceil(worldWidth / cellSize);
    this.cellsPerCol = (int) Math.ceil(worldHeight / cellSize);
    int numCells = cellsPerRow * cellsPerCol;
    dynamicCells = new ArrayList[numCells];
    staticCells = new ArrayList[numCells];
    for (int i = 0; i < numCells; i++) {
        dynamicCells[i] = new ArrayList<>(10);
        staticCells[i] = new ArrayList<>(10);
    }
    foundObjects = new ArrayList<>(10);
}

public void insertStaticObject(Stoixeio obj) {
    int[] cellIds = getCellIds(obj);
    int i = 0;
    int cellId = -1;
    while (i <= 3 && (cellId = cellIds[i++]) != -1) {
        staticCells[cellId].add(obj);
    }
}

public void insertDynamicObject(Stoixeio obj) {
    int[] cellIds = getCellIds(obj);
    int i = 0;
    int cellId = -1;
    while (i <= 3 && (cellId = cellIds[i++]) != -1) {
        dynamicCells[cellId].add(obj);
    }
}

public void removeObject(Stoixeio obj) {
    int[] cellIds = getCellIds(obj);
    int i = 0;
    int cellId = -1;
    while (i <= 3 && (cellId = cellIds[i++]) != -1) {
        dynamicCells[cellId].remove(obj);
        staticCells[cellId].remove(obj);
    }
}

public void clearDynamicCells(Stoixeio obj) {
    int len = dynamicCells.length;
    for (int i = 0; i < len; i++) {
        dynamicCells[i].clear();
    }
}

public ArrayList<Stoixeio> getPotentialColliders(Stoixeio obj) {
    foundObjects.clear();
    int[] cellIds = getCellIds(obj);
    int i = 0;
    int cellId = -1;
    while (i <= 3 && (cellId = cellIds[i++]) != -1) {
        int len = dynamicCells[cellId].size();
        for (int j = 0; j < len; j++) {
            Stoixeio collider = dynamicCells[cellId].get(j);
            if (!foundObjects.contains(collider))
                foundObjects.add(collider);
        }
        len = staticCells[cellId].size();
        for (int j = 0; j < len; j++) {
            Stoixeio collider = staticCells[cellId].get(j);
            if (!foundObjects.contains(collider))
                foundObjects.add(collider);
        }
    }
}

```

```

    }
    return foundObjects;
}

public int[] getCellIds(Stoixeio obj) {
    int x1 = (int)Math.floor(obj.rect.left / cellSize);
    int y1 = (int)Math.floor((obj.rect.bottom + negativeY) / cellSize);
    int x2 = (int)Math.floor((obj.rect.left + obj.rect.width()) / cellSize);
    int y2 = (int)Math.floor((obj.rect.bottom + negativeY + obj.rect.height())
        / cellSize);
    if (x1 == x2 && y1 == y2) {
        if (x1 >= 0 && x1 < cellsPerRow && y1 >= 0 && y1 < cellsPerCol)
            cellIds[0] = x1 + y1 * cellsPerRow;
        else
            cellIds[0] = -1;
            cellIds[1] = -1;
            cellIds[2] = -1;
            cellIds[3] = -1;
    }
    else if (x1 == x2) {
        int i = 0;
        if (x1 >= 0 && x1 < cellsPerRow) {
            if (y1 >= 0 && y1 < cellsPerCol)
                cellIds[i++] = x1 + y1 * cellsPerRow;
            if (y2 >= 0 && y2 < cellsPerCol)
                cellIds[i++] = x1 + y2 * cellsPerRow;
        }
        while (i <= 3) cellIds[i++] = -1;
    }
    else if (y1 == y2) {
        int i = 0;
        if (y1 >= 0 && y1 < cellsPerCol) {
            if (x1 >= 0 && x1 < cellsPerRow)
                cellIds[i++] = x1 + y1 * cellsPerRow;
            if (x2 >= 0 && x2 < cellsPerRow)
                cellIds[i++] = x2 + y1 * cellsPerRow;
        }
        while (i <= 3) cellIds[i++] = -1;
    }
    else {
        int i = 0;
        int y1CellsPerRow = y1 * cellsPerRow;
        int y2CellsPerRow = y2 * cellsPerRow;
        if (x1 >= 0 && x1 < cellsPerRow && y1 >= 0 && y1 < cellsPerCol)
            cellIds[i++] = x1 + y1CellsPerRow;
        if (x2 >= 0 && x2 < cellsPerRow && y1 >= 0 && y1 < cellsPerCol)
            cellIds[i++] = x2 + y1CellsPerRow;
        if (x2 >= 0 && x2 < cellsPerRow && y2 >= 0 && y2 < cellsPerCol)
            cellIds[i++] = x2 + y2CellsPerRow;
        if (x1 >= 0 && x1 < cellsPerRow && y2 >= 0 && y2 < cellsPerCol)
            cellIds[i++] = x1 + y2CellsPerRow;
        while (i <= 3) cellIds[i++] = -1;
    }
    return cellIds;
}
}
}

```

Γ.6 Αρχείο Syndesi.java

```

//*****
// Java Class Name : Syndesi
//*****
package grpekast.sch.users.network;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;

```



```

import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.graphics.Typeface;
import android.os.Bundle;
import android.os.Message;
import android.util.Log;
import android.util.TypedValue;
import android.view.View;
import android.widget.Button;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Set;

//Syndesi Activity
public class Syndesi extends Activity {
    private static int aa = 0;
    public static String REMOTE_DEVICE_ADDRESS = "device_address";
    private static boolean mIsServer = true;
    private BluetoothAdapter mBluetoothAdapter = null;
    private BluetoothDevice bdDevice;
    private ArrayList<BluetoothDevice> arrayListBluetoothDevices = null;
    private ArrayList<BluetoothDevice> arrayListPairedBluetoothDevices;
    private ArrayList<String> arrayListpaired;

    public static boolean isServer() { return mIsServer; }

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

        mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        arrayListpaired = new ArrayList<>();
        arrayListPairedBluetoothDevices = new ArrayList<>();
        arrayListBluetoothDevices = new ArrayList<>();

        Button btnContinue = (Button) findViewById(R.id.serverButton);
        btnContinue.setOnClickListener(onClickSynexeia());
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.i("Syndesi", "onResume() called");
        makeDiscoverable();
        startSearching();
        getPairedDevices();
    }

    @Override
    public synchronized void onResume() {
        super.onResume();
        Log.i("Syndesi", "onResume() called");
    }

    @Override
    public synchronized void onPause() {
        super.onPause();
        Log.i("Syndesi", "onPause() called");
    }

    @Override

```

```

public void onStop() {
    super.onStop();
    Log.i("Syndesi", "onStop() called");
}

@Override
public void onDestroy() {
    super.onDestroy();
    mBluetoothAdapter.cancelDiscovery();
    Syndesi.this.unregisterReceiver(myReceiver);
    aa = 0;
}

private void getPairedDevices() {
    Set<BluetoothDevice> pairedDevice = mBluetoothAdapter.getBondedDevices();
    arrayListpaired.clear();
    arrayListPairedBluetoothDevices.clear();
    if (pairedDevice.size() > 0) {
        for(BluetoothDevice device : pairedDevice) {
            arrayListpaired.add(device.getName() + " " + device.getAddress());
            arrayListPairedBluetoothDevices.add(device);
        }
    }
}

public boolean removeBond(BluetoothDevice btDevice) throws Exception {
    try {
        Method removeBondMethod = btDevice.getClass().getMethod("removeBond",
(Class[]) null);
        return (boolean)removeBondMethod.invoke(btDevice, (Object[])null); }
    catch (Exception e) {e.printStackTrace();
        return false; }
}

public boolean epitygxanetai(BluetoothDevice btDevice) throws Exception {
    try {
        Method createBondMethod = btDevice.getClass().getMethod("createBond",
(Class[]) null);
        return (boolean)createBondMethod.invoke(btDevice, (Object[])null); }
    catch (Exception e) { e.printStackTrace();
        return false; }
}

private void startSearching() {
    Log.i("Log", "in the start searching method");
    IntentFilter intentFilter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
    Syndesi.this.registerReceiver(myReceiver, intentFilter);
    mBluetoothAdapter.startDiscovery();
}

private void makeDiscoverable() {
    if (mBluetoothAdapter.getScanMode() !=
BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE) {
        Intent discoverableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
        discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
300);
        startActivity(discoverableIntent);
    }
    Log.i("Log", "Discoverable ");
}

View.OnClickListener onClickSynexeia() { return new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mBluetoothAdapter.cancelDiscovery();
        setResult(RESULT_CANCELED, new Intent());
        mIsServer = true;
        finish();
    }
}
}

```

```

    };
}

View.OnClickListener onClickSyndesi(final int id) { return new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        // Cancel discovery because it's costly and we're about to connect
        mBluetoothAdapter.cancelDiscovery();
        bdDevice = arrayListBluetoothDevices.get(id);
        Log.i("Log", "Device : " + bdDevice.toString());

        try {
            if (epitygxanetai(bdDevice)) {
                getPairedDevices();
            }
        } catch (Exception e) { e.printStackTrace(); }

        // Get the device MAC address
        String address = bdDevice.getAddress();

        // Create the result Intent and include the MAC address
        Intent intent = new Intent();
        intent.putExtra(REMOTE_DEVICE_ADDRESS, address);
        setResult(RESULT_OK, intent);
        mIsServer = false;
        finish(); //Finish this Activity
    }
};
}

private BroadcastReceiver myReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        Message msg = Message.obtain();
        String action = intent.getAction();

        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);

            // This checks if the size of bluetooth device is 0, then add the device
            if (arrayListBluetoothDevices.size() < 1) // to the arraylist.
                arrayListBluetoothDevices.add(device);
            else {
                for (int i = 0; i < arrayListBluetoothDevices.size(); i++) {
                    if (device.getAddress()
                        .equals(arrayListBluetoothDevices.get(i).getAddress()))
                        return;
                }
                arrayListBluetoothDevices.add(device);
            }
            epiteyxthei(device.getName());
        }
    }
};

private void epiteyxthei(String strDevice) {
    TableLayout table; TableRow tr1, tr2;
    TextView txt, txtNo; Button btn;
    TableRow.LayoutParams params; int px;
    /*if (aa == 0) {
        // Find TextView with title and underline it -----
        txt = (TextView) findViewById((R.id.syndesi_title));
        txt.setPaintFlags(txt.getPaintFlags() | Paint.UNDERLINE_TEXT_FLAG);
    } //-----*/
    aa++;
    // Create LayoutParams -----
    params = new TableRow.LayoutParams(TableRow.LayoutParams.MATCH_PARENT
        , TableRow.LayoutParams.WRAP_CONTENT);
}
}

```

```

px = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 8
    , getResources().getDisplayMetrics());
params.setMargins(0, px, 0, px);
//-----
// Find Tablelayout defined in activity_syndesi.xml -----
table = (TableLayout) findViewById(R.id.syndesi);
//-----
// Create 2 new rows to be added -
tr1 = new TableRow(this);
tr2 = new TableRow(this);
tr1.setLayoutParams(params);
//-----
// Create two TextViews and a Button to be the first row-contents -----
txtNo = new TextView(this);
txt = new TextView(this);
btn = new Button(this);
txtNo.setLayoutParams(params);
txt.setLayoutParams(params);
btn.setLayoutParams(params);
txtNo.setTextSize(TypedValue.COMPLEX_UNIT_SP, 18f);
txt.setTextSize(TypedValue.COMPLEX_UNIT_SP, 18f);
btn.setTextSize(TypedValue.COMPLEX_UNIT_SP, 16f);
txtNo.setText(String.valueOf(aa));
txt.setText(strDevice);
btn.setText("ΣΥΝΔΕΣΗ");
px = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 40
    , getResources().getDisplayMetrics());
txt.setPadding(px, 0, px, 0);
txt.setTypeface(null, Typeface.ITALIC);
btn.setBackgroundResource(R.drawable.syndesi_button);
btn.setOnClickListener(onClickSyndesi(aa-1));
//-----
// Create a Line to be the second row-contents -----
View line = new View(this);
px = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 3
    , getResources().getDisplayMetrics());
params = new TableRow.LayoutParams(TableRow.LayoutParams.MATCH_PARENT, px);
params.span = 3;
line.setLayoutParams(params);
line.setBackgroundColor(0xFF909090);
//-----
// Add TextView, Button to first row and Line to second row.
tr1.addView(txtNo);
tr1.addView(txt);
tr1.addView(btn);
tr2.addView(line);
//-----
// Add rows to TableLayout.
table.addView(tr1);
table.addView(tr2);
//-----
}
}

```

Γ.7 Αρχείο Bluetooth.java

```

//*****
// Java Class Name : Bluetooth
//*****
package grpekast.sch.users.network;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.ObjectInputStream;

```

```

import java.io.ObjectOutputStream;
import java.io.OutputStream;
import java.util.UUID;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;

import grpekast.sch.users.labyrinthos.Platform;

/** This class does all the work for setting up and managing Bluetooth
 * connections with other devices. It has a thread that listens for
 * incoming connections, a thread for connecting with a device, and a
 * thread for performing data transmissions when connected. */
public class Bluetooth {
    // Debugging
    private static final String TAG = "Bluetooth";
    private static final boolean D = true;
    // Name for the SDP record when creating server socket
    private static final String NAME = "Labyrinthos";
    // Unique UUID for this application
    //00001101-0000-1000-8000-00805f9b34fb
    private static final UUID MY_UUID
        = UUID.fromString("78fc5de3-64a1-40ef-bf78-bf190b78afe5");
    // Member fields
    private final BluetoothAdapter mAdapterer;
    private final Handler mHandler;
    private AcceptThread mAcceptThread;
    private ConnectThread mConnectThread;
    private ConnectedThread mConnectedThread;
    private int mState;
    // Constants that indicate the current connection state
    public static final int STATE_NONE = 0; //we're doing nothing
    public static final int STATE_LISTEN = 1; //now listening for incoming connections
    public static final int STATE_CONNECTING = 2; //now initiating an outgoing connection
    public static final int STATE_CONNECTED = 3; //now connected to a remote device

    /** CONSTRUCTOR. Prepares a new BluetoothChat session.
     * @param context The UI Activity Context
     * @param handler A Handler to send messages back to the UI Activity */
    public Bluetooth(Context context, Handler handler) {
        mAdapterer = BluetoothAdapter.getDefaultAdapter();
        mState = STATE_NONE;
        mHandler = handler;
    }

    /** Set the current state of the chat connection
     * @param state An integer defining the current connection state */
    private synchronized void setState(int state) {
        if (D) Log.d(TAG, "setState() " + mState + " -> " + state);
        mState = state;
        // Give the new state to the Handler so the UI Activity can update
        mHandler.obtainMessage(Platform.MESSAGE_STATE_CHANGE, state, -1)
            .sendToTarget();
    }

    /** Return the current connection state. */
    public synchronized int getState() {
        return mState;
    }

    /** Start the chat service. Specifically start AcceptThread to begin a
     * session in listening (server) mode. Called by the Activity onResume() */
    public synchronized void start() {
        if (D) Log.d(TAG, "start");
    }

```

```

// Cancel any thread attempting to make a connection
if (mConnectThread != null) {
    mConnectThread.cancel(); mConnectThread = null;}
// Cancel any thread currently running a connection
if (mConnectedThread != null) {
    mConnectedThread.cancel(); mConnectedThread = null;}
// Start the thread to listen on a BluetoothServerSocket
if (mAcceptThread == null) {
    mAcceptThread = new AcceptThread();
    mAcceptThread.start();
}
setState(STATE_LISTEN);
}

/** Start the ConnectThread to initiate a connection to a remote device.
 * @param device The BluetoothDevice to connect */
public synchronized void connect(BluetoothDevice device) {
    if (D) Log.d(TAG, "connect to: " + device);
    // Cancel any thread attempting to make a connection
    if (mState == STATE_CONNECTING) {
        if (mConnectThread != null) {
            mConnectThread.cancel(); mConnectThread = null;}
    }
    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {
        mConnectedThread.cancel(); mConnectedThread = null;}
    // Start the thread to connect with the given device
    mConnectThread = new ConnectThread(device);
    mConnectThread.start();
    setState(STATE_CONNECTING);
}

/** Start the ConnectedThread to begin managing a Bluetooth connection
 * @param socket The BluetoothSocket on which the connection was made
 * @param device The BluetoothDevice that has been connected */
public synchronized void connected(BluetoothSocket socket, BluetoothDevice device) {
    if (D) Log.d(TAG, "connected");
    // Cancel the thread that completed the connection
    if (mConnectThread != null) {
        mConnectThread.cancel(); mConnectThread = null;}
    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {
        mConnectedThread.cancel(); mConnectedThread = null;}
    // Cancel the accept thread because we only want to connect to one device
    if (mAcceptThread != null) {mAcceptThread.cancel(); mAcceptThread = null;}
    // Start the thread to manage the connection and perform transmissions
    mConnectedThread = new ConnectedThread(socket);
    mConnectedThread.start();
    // Send the name of the connected device back to the UI Activity
    Message msg = mHandler.obtainMessage(Platform.MESSAGE_DEVICE_NAME);
    Bundle bundle = new Bundle();
    bundle.putString(Platform.DEVICE_NAME, device.getName());
    msg.setData(bundle);
    mHandler.sendMessage(msg);
    setState(STATE_CONNECTED);
}

/** Stop all threads */
public synchronized void stop() {
    if (D) Log.d(TAG, "stop");
    if (mConnectThread != null) {
        mConnectThread.cancel(); mConnectThread = null;}
    if (mConnectedThread != null) {
        mConnectedThread.cancel(); mConnectedThread = null;}
    if (mAcceptThread != null) {
        mAcceptThread.cancel(); mAcceptThread = null;}
    setState(STATE_NONE);
}

/** Write to the ConnectedThread in an unsynchronized manner

```

```

* @param obj The bytes to write
* @see ConnectedThread#write(Object) */
public void write(Object obj) {
    // Create temporary object
    ConnectedThread r;
    // Synchronize a copy of the ConnectedThread
    synchronized (this) {
        if (mState != STATE_CONNECTED) return;
        r = mConnectedThread;
    }
    // Perform the write unsynchronized
    r.write(obj);
}

/** Indicate that the connection attempt failed and notify the UI Activity. */
private void connectionFailed() {
    setState(STATE_LISTEN);
    // Send a failure message back to the Activity
    Message msg = mHandler.obtainMessage(Platform.MESSAGE_TOAST);
    Bundle bundle = new Bundle();
    bundle.putString(Platform.TOAST, "Unable to connect device");
    msg.setData(bundle);
    mHandler.sendMessage(msg);
}

/** Indicate that the connection was lost and notify the UI Activity. */
private void connectionLost() {
    setState(STATE_LISTEN);
    // Send a failure message back to the Activity
    Message msg = mHandler.obtainMessage(Platform.MESSAGE_TOAST);
    Bundle bundle = new Bundle();
    bundle.putString(Platform.TOAST, "Device connection was lost");
    msg.setData(bundle);
    mHandler.sendMessage(msg);
}

/** This thread runs while listening for incoming connections. It behaves
 * like a server-side client. It runs until a connection is accepted
 * (or until cancelled). */
private class AcceptThread extends Thread {
    // The local server socket
    private final BluetoothServerSocket mmServerSocket;
    public AcceptThread() {
        BluetoothServerSocket tmp = null;
        // Create a new listening server socket
        try {
            tmp = mAdapter.listenUsingInsecureRfcommWithServiceRecord(NAME, MY_UUID);
        } catch (IOException e) {
            Log.e(TAG, "listen() failed", e);
        }
        mmServerSocket = tmp;
    }
    public void run() {
        if (D) Log.d(TAG, "BEGIN mAcceptThread" + this);
        setName("AcceptThread");
        BluetoothSocket socket = null;
        // Listen to the server socket if we're not connected
        while (mState != STATE_CONNECTED) {
            try {
                // This is a blocking call and will only return on a
                // successful connection or an exception
                socket = mmServerSocket.accept();
            } catch (IOException e) {
                Log.e(TAG, "accept() failed", e);
                break;
            }
            // If a connection was accepted
            if (socket != null) {
                if (D) Log.d(TAG + "mAcceptThread"
                    , "socket != null. A connection was accepted");
            }
        }
    }
}

```

```

        synchronized (Bluetooth.this) {
            switch (mState) {
                case STATE_LISTEN:
                case STATE_CONNECTING:
                    if (D) Log.d(TAG+"mAcceptThread"
                        , "Situation normal. Start the connected thread.");
                    // Situation normal. Start the connected thread.
                    connected(socket, socket.getRemoteDevice());
                    break;
                case STATE_NONE:
                case STATE_CONNECTED:
                    if (D) Log.d(TAG+"mAcceptThread"
                        , "Either not ready or already connected. Terminate new socket.");
                    // Either not ready or already connected. Terminate new socket.
                    try {
                        socket.close();
                        Log.d(TAG + "mAcceptThread", "New socket terminated.");
                    } catch (IOException e) {
                        Log.e(TAG+"mAcceptThread"
                            , "Could not close unwanted socket", e);
                    }
                    break;
            }
        }
    }
    if (D) Log.i(TAG, "END mAcceptThread");
}
public void cancel() {
    if (D) Log.d(TAG, "cancel " + this);
    try {
        mmServerSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of server failed", e);
    }
}
}

/** This thread runs while attempting to make an outgoing connection
 * with a device. It runs straight through; the connection either
 * succeeds or fails. */
private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;
    public ConnectThread(BluetoothDevice device) {
        mmDevice = device;
        BluetoothSocket tmp = null;
        // Get a BluetoothSocket for a connection with the
        // given BluetoothDevice
        try {
            tmp = device.createInsecureRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) {
            Log.e(TAG, "create() failed", e);
        }
        mmSocket = tmp;
    }
    public void run() {
        Log.i(TAG, "BEGIN mConnectThread");
        setName("ConnectThread");
        // Always cancel discovery because it will slow down a connection
        mAdapter.cancelDiscovery();
        // Make a connection to the BluetoothSocket
        try {
            // This is a blocking call and will only return on a
            // successful connection or an exception
            mmSocket.connect();
        } catch (IOException e) {
            connectionFailed();
            // Close the socket
            try {

```



```

        mmSocket.close();
    } catch (IOException e2) {
        Log.e(TAG, "unable to close() socket during connection failure", e2);
    }
    // Start the service over to restart listening mode
    Bluetooth.this.start();
    return;
}
// Reset the ConnectThread because we're done
synchronized (Bluetooth.this) {
    mConnectThread = null;
}
// Start the connected thread
connected(mmSocket, mmDevice);
}
public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}
}

/** This thread runs during a connection with a remote device.
 * It handles all incoming and outgoing transmissions. */
private class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    public ConnectedThread(BluetoothSocket socket) {
        Log.d(TAG, "create ConnectedThread");
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        // Get the BluetoothSocket input and output streams
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) {
            Log.e(TAG, "temp sockets not created", e);
        }
        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run() {
        Log.i(TAG, "BEGIN mConnectedThread");
        final int CHUNK_LENGTH = 1536;
        byte[] buffer, buffercopy;
        int bytes = 0, chunk, pos;
        ByteArrayOutputStream bf = new ByteArrayOutputStream();

        // Keep listening to the InputStream while connected
        while (true) { buffer = new byte[CHUNK_LENGTH];
            try { // Read from the InputStream
                chunk = mmInStream.read(buffer);
                bf.write(buffer);
                bytes += chunk;
                try { Thread.sleep(80); }
                catch (InterruptedException e) {}
                if (chunk == CHUNK_LENGTH) continue;
                buffer = bf.toByteArray();
                bf = new ByteArrayOutputStream();

                buffercopy = new byte[bytes];
                pos = 0;
                while (bytes > 0) {
                    Log.i(TAG, "Bytes remaining: " + bytes);

```

```

        System.arraycopy(buffer, pos, buffercopy, 0, bytes);
        Object obj;
        obj = fromBytes(buffercopy);
        int objsize = toBytes(obj).length;
        pos += objsize;
        bytes -= objsize;
        // Send the obtained bytes to the UI Activity
        mHandler.obtainMessage(Platform.MESSAGE_READ, objsize, -1, obj)
            .sendToTarget();
    }
} catch (IOException e) {
    Log.e(TAG, "disconnected", e);
    connectionLost();
    break;
} catch (ClassNotFoundException e) {
    e.printStackTrace(); }
}
}

/** Write to the connected OutputStream.
 * @param obj The bytes to write */
public void write(Object obj) {
    try {
        byte[] buffer = toBytes(obj);
        mmOutputStream.write(buffer);
        // Share the sent message back to the UI Activity
        mHandler.obtainMessage(Platform.MESSAGE_WRITE, -1, -1, buffer)
            .sendToTarget();
    } catch (IOException e) { Log.e(TAG, "Exception during write", e); }
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}

// Convert object to bytes
private byte[] toBytes(Object obj) throws IOException {
    Log.d(obj.getClass().getName(), "object");
    ByteArrayOutputStream mmByteOut = new ByteArrayOutputStream();
    ObjectOutputStream mmObjectOut = new ObjectOutputStream(mmByteOut);
    mmObjectOut.writeObject(obj);
    mmObjectOut.flush();
    mmObjectOut.reset();
    return mmByteOut.toByteArray();
}

// Restore object from bytes
private Object fromBytes(byte[] bytes)
    throws IOException, ClassNotFoundException {
    ByteArrayInputStream mmByteIn = new ByteArrayInputStream(bytes);
    ObjectInputStream mmObjectIn = new ObjectInputStream(mmByteIn);
    return mmObjectIn.readObject();
}
}
}
}

```

Γ.8 Αρχείο Grafika.java

```

//*****
// Java Class Name : Grafika
//*****
package grpekast.sch.users.utility;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.ColorFilter;

```

```

import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.Point;
import android.graphics.Rect;
import android.graphics.RectF;
import android.graphics.Typeface;
import android.text.StaticLayout;
import android.text.TextPaint;

public class Grafika {
    private Canvas mCanvas;
    private Paint mPaint;
    private TextPaint mTextPaint;
    private Point mPoint;
    private Rect srcRect;
    private Rect dstRect;
    private RectF dstRectF;
    private StaticLayout textbox;
    private Matrix flipHorizontalMatrix;

    //-----
    // Constructor
    //-----
    public Grafika(Canvas myCanvas) {
        mCanvas = myCanvas;
        mPaint = new Paint(Paint.ANTI_ALIAS_FLAG | Paint.LINEAR_TEXT_FLAG);
        mPaint.setStrokeWidth(2);
        mTextPaint = new TextPaint(mPaint);
        mPoint = new Point();
        srcRect = new Rect();
        dstRect = new Rect();
        dstRectF = new RectF();
        flipHorizontalMatrix = new Matrix();
    }

    //-----
    // mergeImages
    //-----
    public static Bitmap mergeImages(Bitmap bmp1, Bitmap bmp2) {
        int width = bmp1.getWidth() > bmp2.getWidth() ?
            bmp1.getWidth() : bmp2.getWidth();
        int height = bmp1.getHeight() > bmp2.getHeight() ?
            bmp1.getHeight() : bmp2.getHeight();
        Bitmap myBitmap = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
        Canvas myCanvas = new Canvas(myBitmap);

        myCanvas.drawBitmap(bmp1
            , (width-bmp1.getWidth())/2, (height-bmp1.getHeight())/2, null);
        myCanvas.drawBitmap(bmp2
            , (width - bmp2.getWidth()) / 2, (height-bmp2.getHeight())/2, null);
        return myBitmap;
    }

    //-----
    // setColor
    //-----
    public void setColor(int myColor) {
        mPaint.setColor(myColor);
    }

    //-----
    // setColorFilter
    //-----
    public void setColorFilter(ColorFilter myFilter) {
        mPaint.setColorFilter(myFilter);
    }
}

```

```

//-----
// setFont
//-----
public void setFont(Typeface myTypeface, float myTextSize) {
    mPaint.setTypeface(myTypeface);
    mPaint.setTextSize(myTextSize);
}

//-----
// setFontDefault
//-----
public void setFontDefault(float myTextSize) {
    mPaint.setTypeface(Typeface.DEFAULT_BOLD);
    mPaint.setTextSize(myTextSize);
}

//-----
// setTextSize
//-----
public void setTextSize(float myTextSize) {
    mPaint.setTextSize(myTextSize);
}

//-----
// getStringWidth
//-----
public int getStringWidth(String str) {
    mPaint.getTextBounds(str, 0, str.length(), dstRect);
    return dstRect.width();
}

//-----
// getStringHeight
//-----
public int getStringHeight(String str) {
    mPaint.getTextBounds(str, 0, str.length(), dstRect);
    return dstRect.height();
}

//-----
// drawString
//-----
public void drawString(String str, int x, int y) {
    mCanvas.drawText(str, x, y, mPaint);
}

//-----
// createTextbox
//-----
public void createTextbox(String str, int width) {
    textbox = new StaticLayout("'" + str, mTextPaint, width,
        StaticLayout.Alignment.ALIGN_NORMAL, 1, 0, false);
}

//-----
// drawTextbox
//-----
public void drawTextbox(int x, int y) {
    mCanvas.translate(x, y);
    textbox.draw(mCanvas);
    mCanvas.translate(-x, -y);
}
public void drawTextbox(String str, int x, int y, int width) {
    createTextbox(str, width);
    drawTextbox(x, y);
}
public void drawTextbox(StaticLayout myTextbox, int x, int y) {
    mCanvas.translate(x, y);
    myTextbox.draw(mCanvas);
    mCanvas.translate(-x, -y);
}

```

```

}

public void drawLine(float x1, float y1, float x2, float y2) {
    mCanvas.drawLine(x1, y1, x2, y2, mPaint);
}

//-----
// fillRect
//-----
public void fillRect(int x, int y, int width, int height) {
    dstRect.set(x, y, x + width, y + height);
    mPaint.setStyle(Paint.Style.FILL);
    mCanvas.drawRect(dstRect, mPaint);
}

//-----
// drawImage
//-----
public void drawImage(Bitmap myBitmap, int x, int y) {
    mCanvas.drawBitmap(myBitmap, x, y, mPaint);
}

public void drawImage(Bitmap myBitmap, int x, int y, int width, int height) {
    srcRect.set(0, 0, myBitmap.getWidth(), myBitmap.getHeight());
    dstRect.set(x, y, x + width, y + height);
    mCanvas.drawBitmap(myBitmap, srcRect, dstRect, mPaint);
}

public void drawImage(Path path) {
    mPaint.setAntiAlias(false);
    mCanvas.drawPath(path, mPaint);
    mPaint.setAntiAlias(true);
}

//-----
// drawFlipImage
//-----
public void drawFlipImage(Bitmap myBitmap, int x, int y, int width, int height) {
    flipHorizontalMatrix.setScale(-1, 1);
    flipHorizontalMatrix.postScale(1f * width / myBitmap.getWidth()
        , 1f * height / myBitmap.getHeight());
    flipHorizontalMatrix.postTranslate(x + width, y);
    mCanvas.drawBitmap(myBitmap, flipHorizontalMatrix, mPaint);
}

//-----
// fillOval
//-----
public void fillOval(float x, float y, float width, float height) {
    dstRectF.set(x, y, x + width, y + height);
    mPaint.setStyle(Paint.Style.FILL);
    mCanvas.drawOval(dstRectF, mPaint);
}

//-----
// scrollScreen
//-----
public void scrollScreen(int dx, int dy) {
    mCanvas.translate((float) dx, (float) dy);
}

//-----
// save / restore / getSaveCount
//-----
public void save() { mCanvas.save(); }
public void restore() { mCanvas.restore(); }
public int getSaveCount() { return mCanvas.getSaveCount(); }

//-----
// clear
//-----

```

```

    public void clear(int color) { mCanvas.drawColor(color);
    }
}

```

Γ.9 Αρχείο Emfanisi.java

```

//*****
// Java Class Name : Emfanisi
//*****
package grpekast.sch.users.utility;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.ColorFilter;
import android.graphics.ColorMatrix;
import android.graphics.ColorMatrixColorFilter;
import android.graphics.Paint;

/*H Emfanisi αφορά τη Σφίγγα*/
public class Emfanisi {
    public static final float COLORFILTER_1 = 60;
    public static final float COLORFILTER_2 = -120;
    public static final float COLORFILTER_3 = 180;
    public static final float COLORFILTER_4 = -45;

    //-----
    // emfanisi[1-4]
    //-----
    public static Bitmap color1(Bitmap myBitmap) {
        return changeHue(myBitmap, COLORFILTER_1); }
    public static Bitmap color2(Bitmap myBitmap) {
        return changeHue(myBitmap, COLORFILTER_2); }
    public static Bitmap color3(Bitmap myBitmap) {
        return changeHue(myBitmap, COLORFILTER_3); }
    public static Bitmap color4(Bitmap myBitmap) {
        return changeHue(myBitmap, COLORFILTER_4); }

    //-----
    // changeHue
    //-----
    private static Bitmap changeHue(Bitmap srcBitmap, Float degrees) {
        ColorFilter colorFilter = adjustHue(degrees);
        Paint paint = new Paint();
        paint.setColorFilter(colorFilter);
        Bitmap myBitmap = Bitmap.createBitmap
            (srcBitmap.getWidth(), srcBitmap.getHeight(), Bitmap.Config.ARGB_8888);
        Canvas canvas = new Canvas(myBitmap);
        canvas.drawBitmap(srcBitmap, 0, 0, paint);
        return myBitmap;
    }

    //-----
    // adjustHue (Creates a HUE adjustment ColorFilter).
    // @param value: degrees to shift the hue.
    //-----
    public static ColorFilter adjustHue( float value ) {
        ColorMatrix cm = new ColorMatrix();
        cm.setSaturation(3f);
        adjustHue(cm, value);
        return new ColorMatrixColorFilter(cm);
    }

    /*@param cm, @param value*/
    public static void adjustHue(ColorMatrix cm, float value) {
        value = cleanValue(value, 180f) / 180f * (float) Math.PI;
        if (value == 0) return;

```

```

float cosVal = (float) Math.cos(value);
float sinVal = (float) Math.sin(value);
float lumR = 0.213f;
float lumG = 0.715f;
float lumB = 0.072f;
float[] mat = new float[] {
    lumR + cosVal * (1 - lumR) + sinVal * (-lumR)
    , lumG + cosVal * (-lumG) + sinVal * (-lumG)
    , lumB + cosVal * (-lumB) + sinVal * (1 - lumB)
    , 0, 0,
    lumR + cosVal * (-lumR) + sinVal * (0.143f)
    , lumG + cosVal * (1 - lumG) + sinVal * (0.140f)
    , lumB + cosVal * (-lumB) + sinVal * (-0.283f)
    , 0, 0,
    lumR + cosVal * (-lumR) + sinVal * (-(1 - lumR))
    , lumG + cosVal * (-lumG) + sinVal * (lumG)
    , lumB + cosVal * (1 - lumB) + sinVal * (lumB)
    , 0, 0,
    0f, 0f, 0f, 1f, 0f,
    0f, 0f, 0f, 0f, 1f };
cm.postConcat(new ColorMatrix(mat));
}

protected static float cleanValue(float p_val, float p_limit) {
    return Math.min(p_limit, Math.max(-p_limit, p_val));
}
}

```

Γ.10 Αρχείο UIButton.java

```

//*****
// Java Class Name : UIButton
//*****
package grpekast.sch.users.utility;

import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.Rect;
import android.text.StaticLayout;
import android.text.TextPaint;

public class UIButton {
    private Rect rectButton; // Declare a Rect object for the button.
    private boolean buttonDown = false; // Determine whether the button is pressed down.
    private Bitmap bmpButton, bmpButtonDown; // Declare images for the button.
    private StaticLayout txtButton; // Declare textbox for the button.
    private int bgcolor, bgcolorDown; // Declare background colors for the textbox.
    private boolean isText = false; // Declare if button is text for rendering.

    //-----
    // Constructors
    //-----
    public UIButton(int left, int top, int right, int bottom
        , Bitmap bmpButton, Bitmap bmpButtonDown) {
        rectButton = new Rect(left, top, right, bottom);
        this.bmpButton = bmpButton;
        this.bmpButtonDown = bmpButtonDown;
    }
    public UIButton(int left, int top, int right, int bottom, Bitmap bmpButton) {
        int width = bmpButton.getWidth(), height = bmpButton.getHeight();
        Bitmap bmpButtonDown = Bitmap.createBitmap(width,height, Bitmap.Config.RGB_565);
        bmpButtonDown.eraseColor(Color.rgb(200, 200, 175));

        rectButton = new Rect(left, top, right, bottom);
        this.bmpButton = bmpButton;
    }
}

```

```

        this.bmpButtonDown = Grafika.mergeImages bmpButtonDown
            , Bitmap.createScaledBitmap bmpButton, width-4, height-4, false));
    }
    public UIButton(int left, int top, int width, String str
        , int size, int color, int bgcolor, int bgcolorDown) {
        TextPaint tp = new TextPaint();
        tp.setAntiAlias(true); tp.setLinearText(true);
        tp.setTextSize(size);
        tp.setColor(color); this.bgcolor = bgcolor; this.bgcolorDown = bgcolorDown;
        txtButton = new StaticLayout(" " + str, tp, width,
            StaticLayout.Alignment.ALIGN_NORMAL, 1, 0, false);
        rectButton = new Rect(left, top, left + width, top + txtButton.getHeight());
        isText = true;
    }

    public void render(Grafika g) {
        if (isText) { // Draw the Textbox Button.
            if (buttonDown) g.setColor(bgcolorDown); else g.setColor(bgcolor);
            g.fillRect(rectButton.left, rectButton.top
                , rectButton.width(), rectButton.height());
            g.drawTextbox(txtButton, rectButton.left, rectButton.top);
        }
        else { //Draw the Button Image.
            Bitmap currentButtonImage = buttonDown ? bmpButtonDown : bmpButton;
            g.drawImage(currentButtonImage, rectButton.left, rectButton.top
                , rectButton.width(), rectButton.height());
        }
    }

    public void onTouchDown(int touchX, int touchY) {
        if (rectButton.contains(touchX, touchY)) {
            buttonDown = true;
        } else {
            buttonDown = false;
        }
    }

    public void onSelect(int touchX, int touchY) {
        if (rectButton.contains(touchX, touchY)) buttonDown = !buttonDown;
    }

    public void cancel() {
        //Button has been released.
        buttonDown = false;
    }

    public boolean isPressed(int touchX, int touchY) {
        //Check If the button is active and the release was within.
        return buttonDown && rectButton.contains(touchX, touchY);
    }

    public boolean isSelected() {
        //Check If the button is active.
        return buttonDown;
    }

    public int getX() { return rectButton.left; }
    public int getY() { return rectButton.top; }
    public int getHeight() { return rectButton.height(); }
}

```

Γ.11 Αρχείο TouchScreen.java

```

//*****
// Java Class Name : TouchScreen
//*****

```



```

package grpekast.sch.users.utility;

import android.view.MotionEvent;
import android.view.View;

import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.state.State;

public class TouchScreen implements View.OnTouchListener {
    private State currentState;

    //-----
    // setCurrentState
    //-----
    public void setCurrentState(State currentState) {
        this.currentState = currentState; }

    //-----
    // onTouch
    @Override //-----
    public boolean onTouch(View v, MotionEvent event) {
        int scaledX = (int) ((event.getX() / v.getWidth()) * Platform.GAME_WIDTH);
        int scaledY = (int) ((event.getY() / v.getHeight()) * Platform.GAME_HEIGHT);
        boolean eventHandled = currentState.onTouch(event, scaledX, scaledY);
        if (!currentState.isThreaded && eventHandled) Platform.sGame.invalidate();
        return eventHandled;
    }
}

```

Γ.12 Αρχείο Accelerometer.java

```

//*****
// Java Class Name : Accelerometer
//*****
package grpekast.sch.users.utility;

import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.view.Surface;

import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.state.State;

public class Accelerometer implements SensorEventListener {
    private SensorManager sensorMgr;
    private State currentState;

    //-----
    // Constructor
    //-----
    public Accelerometer(Context context) {
        sensorMgr = (SensorManager) context.getSystemService(Context.SENSOR_SERVICE);
        if (sensorMgr.getSensorList(Sensor.TYPE_ACCELEROMETER).size() != 0) {
            Sensor accelerometer = sensorMgr
                .getSensorList(Sensor.TYPE_ACCELEROMETER).get(0);
            sensorMgr.registerListener(
                this, accelerometer, SensorManager.SENSOR_DELAY_GAME);
        }
    }
}

```

```

//-----
// setCurrentState
//-----
public void setCurrentState(State currentState) {
    this.currentState = currentState; }

public void onAccuracyChanged(Sensor sensor, int accuracy) {}

//-----
// onSensorChanged
@Override //-----
public void onSensorChanged(SensorEvent event) {
    float accelX;
    float accelY;
    if (Platform.iRotation == Surface.ROTATION_90) {
        accelX = event.values[0];
        accelY = event.values[1];
    }
    else { //Surface.ROTATION_0
        accelX = event.values[1];
        accelY = -event.values[0];
    }
    currentState.onSensor(accelX, accelY);
}

//-----
// stopSensor
//-----
public void stopSensor() { sensorMgr.unregisterListener(this); }
}

```

Γ.13 Αρχείο Basi.java

```

//*****
// Java Class Name : Basi
//*****
package grpekast.sch.users.database;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Environment;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import grpekast.sch.users.labyrinthos.Platform;

public class Basi extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "enigmadb"; // database name
    public static final int DATABASE_VERSION = 1; // database version
    public static String DATABASE_PATH;
    public static final String TABLE_1 = "CATEGORIES";
    public static final String TABLE_2 = "EROTAPANTISI4";
    public static final String TABLE_3 = "EROTAPANTISI9";
    public static final String FIELD_ID = "ID";
    public static final String FIELD_CATEGORY_ID = "CATEGID";
    public static final String FIELD_CATEGORY = "CATEGORY";
    public static final String FIELD_LEVEL = "EPIPEDO";
    public static final String FIELD_THEME = "THEMA";
}

```

```

public static final String FIELD_CORRECT = "SOSTO";
public static final String FIELD_WRONG = "LATHOS";
private static String[] categories;
private SQLiteDatabase mDatabase;
private int[][] maxIdTable;

//-----
// Constructor
//-----
public Basi(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
    String dbPath = context.getDatabasePath(DATABASE_NAME).toString();
    DATABASE_PATH = dbPath.substring(0, dbPath.length() - DATABASE_NAME.length());

    try { createDataBase(); } //If database does not exist copy it from the assets
    catch (IOException e) { throw new Error("ErrorCopyingDataBase"); }

    readCategories();
}

@Override
public void onCreate(SQLiteDatabase db) { /*createTables(db);*/ }
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {}

//-----
// close
@Override //-----
public synchronized void close() {
    if(mDatabase != null)
        mDatabase.close();
    super.close();
}

//-----
// createTables
//-----
private void createTables(SQLiteDatabase db) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_1 + ";" );
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_2 + ";" );
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_3 + ";" );
    // SQL statements to create enigma tables
    String CREATE_CATEGORIES_TABLE =
        "CREATE TABLE IF NOT EXISTS " + TABLE_1 + " ( "
        + FIELD_ID + " BYTE PRIMARY KEY, "
        + FIELD_CATEGORY + " TEXT)";
    String CREATE_EROTAPANTISI4_TABLE =
        "CREATE TABLE IF NOT EXISTS " + TABLE_2 + " ( "
        + FIELD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + FIELD_CATEGORY_ID + " BYTE, "
        + FIELD_LEVEL + " BYTE, "
        + FIELD_THEME + " TEXT, "
        + FIELD_CORRECT + "1 TEXT";
    String CREATE_EROTAPANTISI9_TABLE =
        "CREATE TABLE IF NOT EXISTS " + TABLE_3 + " ( "
        + FIELD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + FIELD_CATEGORY_ID + " BYTE, "
        + FIELD_LEVEL + " BYTE, "
        + FIELD_THEME + " TEXT";

    int i;
    for (i = 1; i <= 4; i++)
        CREATE_EROTAPANTISI9_TABLE += ", " + FIELD_CORRECT + i + " TEXT";
    for (i = 1; i <= 3; i++)
        CREATE_EROTAPANTISI4_TABLE += ", " + FIELD_WRONG + i + " TEXT";
    for (i = 1; i <= 5; i++)
        CREATE_EROTAPANTISI9_TABLE += ", " + FIELD_WRONG + i + " TEXT";
    CREATE_EROTAPANTISI4_TABLE += " )";
    CREATE_EROTAPANTISI9_TABLE += " )";

    db.execSQL(CREATE_EROTAPANTISI4_TABLE);
}

```

```

db.execSQL(CREATE_EROTAPANTISI9_TABLE);
db.execSQL(CREATE_CATEGORIES_TABLE);
// Create index for tables 2 and 3.
db.execSQL("CREATE INDEX CINDEK2 ON " + TABLE_2 +
    " (" + FIELD_CATEGORY_ID + ", " + FIELD_LEVEL + ");");
db.execSQL("CREATE INDEX CINDEK3 ON " + TABLE_3 +
    " (" + FIELD_CATEGORY_ID + ", " + FIELD_LEVEL + ");");
}

//-----
// createDataBase
//-----
public void createDataBase() throws IOException {
    //If database does not exist copy it from the assets
    boolean mDataBaseExist = checkDataBase();
    if(!mDataBaseExist) {
        this.getReadableDatabase();
        this.close();
        try { //Copy the database from assets
            loadDefaultDatabase();
            exportDB();
            System.out.println("database created");
        } catch (IOException e) { throw new Error("ErrorCopyingDataBase"); }
    }
    mDatabase = getWritableDatabase();
}

//-----
// checkDataBase
//-----
/**Check if the database already exists
 * @return true if it exists, false if it doesn't */
private boolean checkDataBase() {
    File dbFile = new File(DATABASE_PATH + DATABASE_NAME);
    return dbFile.exists();
}

//-----
// loadDefaultDatabase
//-----
/**Copies the database from the local assets-folder to the system folder,
 * where it can be accessed and handled.
 * This is done by transferring bytestream. */
private static void loadDefaultDatabase() throws IOException {
    // Path to the to be created db
    String outFile_name = DATABASE_PATH + DATABASE_NAME;
    //Open the local db as the input stream
    InputStream inputStream = Platform.assets.open("enigmadb");
    //Open the empty db as the output stream
    OutputStream outputStream = new FileOutputStream(outFile_name);
    //transfer bytes from the inputfile to the outputfile
    byte[] buffer = new byte[1024];
    int length;
    while ((length = inputStream.read(buffer))>0) {
        outputStream.write(buffer, 0, length);
    }
    //Close the streams
    outputStream.flush();
    outputStream.close();
    inputStream.close();
}

//-----
// exportDB
//-----
/**Copies the database from the local system data folder to sd card.
 * This is done by transferring bytestream. */
private void exportDB() {
    try { File sd = Environment.getExternalStorageDirectory();
        if (sd.canWrite()) {

```

```

        // Local database
        InputStream input= new FileInputStream(getWritableDatabase().getPath());
        // create directory for backup
        File backupDir = new File(sd.getPath() + DATABASE_PATH);
        if(!backupDir.mkdirs()) System.out.println("Directory already exists.");
        // Path to the external backup
        OutputStream output =
            new FileOutputStream(backupDir + File.separator + DATABASE_NAME);
        // transfer bytes from the Input File to the Output File
        byte[] buffer = new byte[1024];
        int length;
        while ((length = input.read(buffer)) > 0)
            output.write(buffer, 0, length);
        output.flush();
        output.close();
        input.close();
    }
} catch (Exception e) { System.out.println("Export Failed!"); }
}

//-----
// deleteDB
//-----
/**Deletes a database file from local system data folder.*/
private void deleteDB(String dbName) {
    File dbFile =new File(DATABASE_PATH + dbName);
    if (dbFile.exists()) {
        System.out.println("File '" + dbName + "' found.\nDeleting...");
        if (dbFile.delete()) System.out.println("File deleted.");
        else System.out.println("Delete failed!");
    }
    else System.out.println("File '" + dbName + "' not found!");
}

//-----
// readCategories
//-----
private boolean readCategories() {
    Cursor rst, rstCategories = mDatabase.rawQuery(
        "SELECT " + FIELD_ID + ", " + FIELD_CATEGORY +
        " FROM " + TABLE_1 + " ORDER BY " + FIELD_ID, null);
    int count = rstCategories.getCount();
    if (count == 0)
        { rstCategories.close(); return false; }
    maxIdTable = new int[2][count+1];
    categories = new String[count];
    String categid;
    rstCategories.moveToFirst();
    for (int i = 1; i <= count; i++) {
        categid = String.valueOf(rstCategories.getInt(0));
        categories[i-1] = rstCategories.isNull(1)? "": rstCategories.getString(1);
        rst = mDatabase.rawQuery(
            "SELECT " + FIELD_ID + " FROM " + TABLE_2 +
            " WHERE " + FIELD_CATEGORY_ID + "=? ORDER BY " + FIELD_ID +
            " DESC LIMIT 1", new String[]{categid});
        maxIdTable[0][i] = rst.moveToFirst() ? rst.getInt(0) : 0;
        rst.close();
        rst = mDatabase.rawQuery(
            "SELECT " + FIELD_ID + " FROM " + TABLE_3 +
            " WHERE " + FIELD_CATEGORY_ID + "=? ORDER BY " + FIELD_ID +
            " DESC LIMIT 1", new String[]{categid});
        maxIdTable[1][i] = rst.moveToFirst() ? rst.getInt(0) : 0;
        rst.close();
        rstCategories.moveToNext();
    }
    rstCategories.close();
    return true;
}

//-----

```

```

// getCategoryes
//-----
public static String getCategoryes(int index) { return categories[index]; }

//-----
// getDB
//-----
public SQLiteDatabase getDB() { return mDatabase; }

//-----
// getMaxId
//-----
public int getMaxId(String table, byte categ) {
    return table.equals(TABLE_2) ? maxIdTable[0][categ] : maxIdTable[1][categ];
}
}

```

Γ.14 Αρχείο Enigma.java

```

//*****
// Java Class Name : Enigma
//*****
package grpekast.sch.users.database;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import java.io.Serializable;
import java.util.Random;

import grpekast.sch.users.labyrinthos.Game;
import grpekast.sch.users.labyrinthos.Platform;

public class Enigma implements Serializable {
    public final Erotisi erotisi;
    public final Apantisi[] apantisi;
    private int id; //tyxaia - rand.nextInt(maxId)+1 (id)
    private transient Basi mBasi;
    private Random rand = new Random();
    private static final long serialVersionUID = 455530706921004893L;

    //-----
    // Constructor
    //-----
    public Enigma(byte type, byte category, byte level) {
        Platform.sGame.openDBConnection(this);
        category++;
        apantisi = new Apantisi[type];
        erotisi = new Erotisi(readEnigma(type, category, level));
        erotisi.setCategory(readCategory(category));
    }

    //-----
    // readCategory
    //-----
    private String readCategory(byte category) {
        SQLiteDatabase db = mBasi.getDB();
        Cursor rstCategories = db.rawQuery(
            "SELECT " + Basi.FIELD_CATEGORY + " FROM " + Basi.TABLE_1 +
            " WHERE " + Basi.FIELD_ID + "=?",
            new String[]{String.valueOf(category)});
        int column = rstCategories.getColumnIndex(Basi.FIELD_CATEGORY);
        String str = rstCategories.moveToFirst() ? rstCategories.getString(column) : "";
        rstCategories.close();
        return str;
    }
}

```

```

}

//-----
// readEnigma
//-----
private String readEnigma(byte type, byte category, byte level) {
    String strTable = "EROTAPANTISI" + type;
    String strQuestion, strNull = "--- KAMIA ---";
    int i, column, maxId = mBasi.getMaxId(strTable, category);
    SQLiteDatabase db = mBasi.getDB();

    Cursor rst = db.rawQuery("SELECT * FROM " + strTable +
        " WHERE (ID=? OR ID>=?) AND CATEGID=? AND EPIPEDO=? ORDER BY ID LIMIT 1"
        , new String[]{String.valueOf(maxId)
            , String.valueOf(rand.nextInt(maxId) + 1)
            , String.valueOf(category), String.valueOf(level)});
    id = rst.moveToFirst() ? rst.getInt(0) : 0;
    if (id == 0) { //επανερώτηση εάν δεν υπάρχει το αιτούμενο επίπεδο δυσκολίας
        rst = db.rawQuery("SELECT * FROM " + strTable +
            " WHERE CATEGID=? AND EPIPEDO<=? ORDER BY EPIPEDO DESC LIMIT 1"
            , new String[]{String.valueOf(category), String.valueOf(level)});
        id = rst.moveToFirst() ? rst.getInt(0) : 0;
    }

    column = rst.getColumnIndex(Basi.FIELD_THEME);
    strQuestion = id==0 || rst.isNull(column) ? strNull : rst.getString(column);

    int count = (type == 4 ? 1 : 4);
    for (i = 0; i < count; i++) {
        column = rst.getColumnIndex(Basi.FIELD_CORRECT + (i+1));
        apantisi[i] = new Apantisi(id==0 || rst.isNull(column) ? strNull :
            rst.getString(column), true);
    }
    count = (type == 4 ? 3 : 5);
    for (int j = 1; j <= count; i++, j++) {
        column = rst.getColumnIndex(Basi.FIELD_WRONG + j);
        apantisi[i] = new Apantisi(id==0 || rst.isNull(column) ? strNull :
            rst.getString(column), false);
    }
    rst.close();
    return strQuestion;
}

public int getId() { return id; }
public void setId(int id) { this.id = id; }

//-----
// receiveBasi
//-----
public void receiveBasi(Game.DBConnection conn) { mBasi = conn.getDBHelper(); }
}

```

Γ.15 Αρχείο Erotisi.java

```

//*****
// Java Class Name : Erotisi
//*****
package grpekast.sch.users.database;

import java.io.Serializable;

public class Erotisi implements Serializable {
    public final String string;
    private String category;
    private static final long serialVersionUID = 455530706921004894L;
}

```

```

//-----
// Constructor
//-----
public Erotisi(String str) {
    string = str;
}

public boolean equals(String str) { return string.equals(str); }

//-----
// setCategory
//-----
public void setCategory(String str) { category = str; }

//-----
// getCategory
//-----
public String getCategory() { return category; }
}

```

Γ.16 Αρχείο Apantisi.java

```

//*****
// Java Class Name : Apantisi
//*****
package grpekast.sch.users.database;

import java.io.Serializable;

public class Apantisi implements Serializable {
    public final String string;
    public final boolean sosto;
    public boolean epilogi;
    private byte count = 0;
    private static final long serialVersionUID = 455530706921004895L;

    //-----
    // Constructor
    //-----
    public Apantisi(String str, boolean sosto) {
        string = str;
        this.sosto = sosto;
        count++;
    }

    public void provaletai() { }
}

```

Γ.17 Αρχείο Stoixeio.java

```

//*****
// Java Class Name : Stoixeio
//*****
package grpekast.sch.users.model;

import android.graphics.Point;
import android.graphics.PointF;
import android.graphics.RectF;

import java.io.ObjectStreamException;

```



```

import java.io.Serializable;

public class Stoixeio implements Serializable {
    public final transient Point grid;
    public final transient PointF position;
    public final transient RectF rect;
    private static final long serialVersionUID = 455530706921004900L;

    //-----
    // Constructor
    //-----
    public Stoixeio(float x, float y, float width, float height) {
        this.position = new PointF(x, y);
        this.rect = new RectF(x - width/2, y - height/2, x + width/2, y + height/2);
        this.grid = new Point();
    }

    //-----
    // Copy Constructor
    //-----
    protected Stoixeio(Stoixeio stoixeio) {
        this.position = new PointF();
        this.rect = new RectF();
        this.grid = new Point();
    } // create a new object from the deserialized one
    protected Object readResolve() throws ObjectStreamException {
        return new Stoixeio(this);
    }

    //-----
    // reconstruct
    //-----
    public void reconstruct(float x, float y, float width, float height) {
        this.position.set(x, y);
        this.rect.set(x - width/2, y - height/2, x + width/2, y + height/2);
    }
}

```

Γ.18 Αρχείο Player.java

```

//*****
// Java Class Name : Player
//*****
package grpekast.sch.users.model;

import android.graphics.Point;
import android.graphics.Rect;

import grpekast.sch.users.labyrinthos.Platform;

public class Player extends Stoixeio {
    private static byte count = 0;
    public static final byte aa = count;
    public final int icon;
    public boolean multi;
    protected boolean sigouros;
    protected boolean ikanos;
    public boolean isMoving = false;
    public enum Fasi { LABYRINTHOS, BONUS }
    protected Fasi fasi = Fasi.LABYRINTHOS;
    private static final int U = 50; //U: Velocity
    private final byte WALK_MAX = 2;
    private byte walk = 0;
    private byte pushBackCount = 0;
}

```

```

private byte lymenaEnigmata = 0;
private byte apantiseis = 0;
private final int width, height;
private int x, y;
private int backupX, backupY;
private boolean moveOnAxisX = true;
private final Point newpos;
private final Rect irect;

//-----
// Constructor
//-----
public Player(int x, int y, int width, int height, int index) {
    super(x, y, width, height);
    this.x = x;
    this.y = y;
    this.width = width;
    this.height = height;
    backupX = x;
    backupY = (Platform.GAME_HEIGHT - height) / 2;
    irect.set(x, y, x+width, y+height);
    irect = new Rect(x, y, x+width, y+height);
    newpos = new Point(x, y);
    icon = index;
    ikanos = false;
    count++;
}

//-----
// move
//-----
public boolean move(float dt) {
    if (dt == 0) return isMoving = true;
    if (x != backupX || y != backupY) pushBackCount = 0;
    backupX = x; backupY = y;

    if (moveOnAxisX) {
        x += (newpos.x >= x ? U * dt : -U * dt);
    } else {
        y += (newpos.y >= y ? U * dt : -U * dt);
    }
    irect.offsetTo(x, y);
    position.set(irect.centerX(), irect.centerY());

    if (irect.contains(moveOnAxisX ? newpos.x : x, moveOnAxisX ? y : newpos.y)) {
        if (walk < WALK_MAX) {
            moveOnAxisX = !moveOnAxisX;
            walk++;
        } else return isMoving = false;
    }
    return isMoving = true;
}

public boolean move(int dx, int dy) {
    backupX = x; backupY = y;
    x += dx; y += dy;
    irect.offsetTo(x, y);
    position.x += dx; position.y += dy;
    if (irect.contains(newpos.x, newpos.y)) return isMoving = false;
    else return isMoving = true;
}

//-----
// pushBack
//-----
public byte pushBack() {
    x = backupX; y = backupY;
    irect.offsetTo(x, y);
    position.x = irect.centerX(); position.y = irect.centerY();

    if ((Math.abs(x-newpos.x) > width/2) && (Math.abs(y-newpos.y) > height/2)) {

```

```

        moveOnAxisX = !moveOnAxisX;
    }
    return ++pushBackCount;
}

//-----
// topotheteitai
//-----
public void topotheteitai(int x, int y) {
    backupX = this.x; backupY = this.y;
    moveOnAxisX = Math.abs(x - this.x - width/2) > Math.abs(y - this.y - height/2);
    newPos.set(x + (x > this.x + width/2 ? width : -width)/2 ,
                y + (y > this.y + height/2 ? height : -height)/2);
    walk = ((Math.abs(x-backupX) < width/2) || (Math.abs(y-backupY) < height/2))
            ? WALK_MAX : 1;
    pushBackCount = 0;
}

//-----
// eiserxetai
//-----
public void eiserxetai(Fasi myFasi) {
    lymenaEnigmata = 0;
    apantiseis = 0;
    ikanos = false;
    fasi = myFasi;
}

public void pefteiSeSfigga() {
    pushBack();
    isMoving = true;
}

public boolean pernaei() {
    return false;
}

protected void synantaei() {}

protected void pernaeiErotisi() {}

protected int epilegei() { return apantiseis; }

//-----
// apantaei
//-----
protected int apantaei() {
    isMoving = false;
    return ++apantiseis;
}

//-----
// lynei
//-----
public byte lynei() {
    apantaei();
    return ++lymenaEnigmata;
}

//-----
// synexizei
//-----
public boolean synexizei() { return ikanos; }
public void synexizei(boolean qualified) { this.ikanos = qualified; }

//-----
// set
//-----
public void set(int x, int y) {
    this.x = x;
}

```

```

        this.y = y;
        backupX = x;
        backupY = (Platform.GAME_HEIGHT - height) / 2;
        rect.offsetTo(x, y);
        irect.offsetTo(x, y);
        position.set(rect.centerX(), rect.centerY());
        newpos.set(irect.centerX(), irect.centerY());
    }

    //-----
    // getX
    //-----
    public int getX() { return x; }

    //-----
    // getY
    //-----
    public int getY() { return y; }

    //-----
    // getWidth
    //-----
    public int getWidth() { return width; }

    //-----
    // getHeight
    //-----
    public int getHeight() { return height; }

    //-----
    // getOldX
    //-----
    public int getOldX() { return backupX; }

    //-----
    // getOldY
    //-----
    public int getOldY() { return backupY; }

    //-----
    // getRect
    //-----
    public Rect getRect() { rect.round(irect); return irect; }

    //-----
    // getLymenaEnigmata
    //-----
    public byte getLymenaEnigmata() { return lymenaEnigmata; }
}

```

Γ.19 Αρχείο Sfigga.java

```

//*****
// Java Class Name : Sfigga
//*****
package grpekast.sch.users.model;

import java.io.ObjectStreamException;
import java.io.Serializable;
import java.util.Random;

import grpekast.sch.users.database.Enigma;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.state.Eidos;
import grpekast.sch.users.state.Labyrinth;
import grpekast.sch.users.state.MultipleChoice;

```

```

import grpekast.sch.users.state.Pinax;
import grpekast.sch.users.state.State;

public class Sfigga extends Stoixeio implements Serializable {
    public final int left, top;
    public final byte category, type;
    private Enigma enigma;
    private transient Player player;
    private transient Eidos eidos;
    private int apotelesma = 0;
    private byte labyrNo = 0;
    private static final long serialVersionUID = 455530706921004901L;

    //-----
    // Constructors
    //-----
    public Sfigga(byte type, byte category, byte level) {
        super(0, 0, 0, 0);
        left = 0; top = 0;
        this.category = category;
        this.type = type;
        enigma = new Enigma(type, category, level);
    }
    public Sfigga(float x, float y, float width, float height,
        byte category, byte level) {
        super(x, y, width, height);
        left = (int)rect.left; top = (int)rect.top;
        rect.set(rect.left+3, rect.top, rect.right-3, rect.bottom);

        this.category = category;
        type = (byte) ((new Random().nextInt(9) <= 5 ? 4 : 9));
        enigma = new Enigma(type, category, level);
    }
    //-----
    // Copy Constructor
    //-----
    private Sfigga(Sfigga sfigga) {
        super(sfigga);
        left = sfigga.left; top = sfigga.top;
        category = sfigga.category; type = sfigga.type;
        enigma = sfigga.enigma;
    }
    @Override // create a new object from the deserialized one
    protected Object readResolve() throws ObjectStreamException {
        return new Sfigga(this);
    }
    @Override
    public void reconstruct(float x, float y, float width, float height) {
        super.reconstruct(x, y, width, height);
        rect.left = left + 3;
        rect.right = rect.right - 3;
    }

    public void emfanizetai() {} // Labyrinth -> createSfigga

    //-----
    // kaneiErotisi
    //-----
    public Eidos kaneiErotisi(State state, Player player) {
        this.player = player;

        if (state instanceof Labyrinth) {
            Labyrinth labyrinth = (Labyrinth) state;
            labyrNo = labyrinth.getLabyrNo();
            switch (type) {
                case 9: eidos = new Pinax(labyrinth, labyrinth.currentClip); break;
                default: eidos = new MultipleChoice(labyrinth, labyrinth.currentClip);
            }
        } else
    }
}

```

```

        switch (type) {
            case 9: eidos = new Pinax(state); break;
            default: eidos = new MultipleChoice(state);
        }
        eidos.setCategory(category);
        eidos.provalei(this, enigma);
        return eidos;
    }

    //-----
    // lysiEnigmatos
    //-----
    public void lysiEnigmatos(int failedAttempt) {
        byte n;
        apotelesma = 10 - failedAttempt*5;

        switch (player.fasi) {
            case LABYRINTHOS:
                if (apotelesma <= 0) {
                    player.apantaei();
                    Platform.vibrator.vibrate(300);
                    eidos.setNotification("Έχασες 1 ζωή.");
                } else if ((n = player.lynei()) >= 3) {
                    eidos.setNotification(labyrNo > 1 ?
                        "Η πόρτα άνοιξε!" : "Το σεντούκι άνοιξε!");
                    player.synexizei(true);
                } else
                    eidos.setNotification("Μένουν αινίγματα: " + (3 - n));
                break;
            case BONUS:
                if (apotelesma > 0) {
                    player.lynei();
                    eidos.setNotification("Κέρδισες " + (type == 4 ?
                        1 + " ζωή!" : 2 + " ζωές!");
                } else if (player.apantaei() < 2) {
                    Platform.vibrator.vibrate(300);
                    eidos.setNotification("Έχασες 1 ζωή!");
                }
                break;
            default: if (apotelesma > 0) {
                player.lynei();
                eidos.setNotification("Μπράβο!");
            }
        }
    }

    //-----
    // getApotelesma
    //-----
    public int getApotelesma() { return apotelesma; }
}

```

Γ.20 Αρχείο Diadromi.java

```

//*****
// Java Class Name : Diadromi
//*****
package grpekast.sch.users.model;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Random;

import static grpekast.sch.users.model.Diadromi.Way.*;

public class Diadromi implements Serializable {
    //----- Enumerate type -----|

```

```

public enum Way {NONE, LEFT, RIGHT, UP, DOWN, LEFTRIGHT, UPDOWN} //|
//-----|
public final int MHKOS;
private static Random rand = new Random();
private Way path[];
private static final long serialVersionUID = -455530706921004893L;

//-----
// Constructor
//-----
public Diadromi(int lengthAxisX) { MHKOS = dimiourgeitai(lengthAxisX); }

//-----
// getPath
//-----
public Way[] getPath() { return path; }

//-----
// dimiourgeitai
//-----
private int dimiourgeitai(int lengthX) {
    ArrayList<Way> wayList = new ArrayList<>();
    Way way;
    int lowerBound = Way.RIGHT.ordinal();
    int upperBound = Way.LEFTRIGHT.ordinal() + 1;
    int i = 1;

    wayList.add(RIGHT);
    while (i < lengthX) {
        way = Way.values()[rand.nextInt(upperBound-lowerBound)+lowerBound];
        if (way == LEFTRIGHT) way = RIGHT;
        if (way == RIGHT) i++;
        else if (wayList.lastIndexOf(RIGHT) != wayList.size()-1) {
            if (way != wayList.get(wayList.size()-1)) continue;
        }
        wayList.add(way);
    } wayList.add(RIGHT);

    path = new Way[wayList.size()];
    wayList.toArray(path);
    return wayList.size();
}
}

```

Γ.21 Αρχείο Eisodos.java

```

//*****
// Java Class Name : Eisodos
//*****
package grpekast.sch.users.model;

import android.graphics.Point;

import java.util.ArrayList;

public class Eisodos {
    public Point pos;
    public final byte id;
    public static ArrayList<Eisodos> eisodosList = new ArrayList<>();
    private static byte count = 0;
    private int breadth;

    //-----
    // Constructors
    //-----
    public Eisodos(int y1, int y2, int breadth) {

```

```

        this.breadth = breadth;
        createEisodos(y1, y2);
        count++;
        id = count;
    }
    public Eisodos(int dy) {
        int y0;
        eisodosList.add(this);
        count++;
        y0 = (dy % (2*count + 1)) / 2;
        breadth = (dy - 2*y0) / (2*count + 1);
        updateEisodosList(y0);
        id = count;
    }

    //-----
    // init
    //-----
    public static void init() {
        count = 0;
        eisodosList.clear();
    }

    //-----
    // createEisodos
    //-----
    private void createEisodos(int y1, int y2) {
        int y0, midLength = 0, dy = Math.abs(y1 - y2);
        byte groupCount = 1;
        if (dy == breadth) breadth -= 10;

        for (Eisodos eisodos : eisodosList)
        {   if (eisodos.pos.y >= y1 && eisodos.pos.y < y2) groupCount++; }
        if (groupCount > 1)
            midLength = breadth * (int) ((dy/breadth - groupCount)/(groupCount-1));
        y0 = y1 + (dy - (groupCount*breadth + (groupCount-1)*midLength)) / 2;

        if (eisodosList.size() > 0) updateEisodosList(y1, y2, y0, midLength);
        pos = new Point(0, y0 + (groupCount-1)*(breadth + midLength));
        eisodosList.add(this);
    }

    //-----
    // updateEisodosList
    //-----
    private void updateEisodosList(int y0) {
        for (int i = 1; i <= count; i++) {
            eisodosList.get(i - 1).pos = new Point(0, y0 + (2*i - 1)*breadth);
            eisodosList.get(i - 1).breadth = breadth;
        }
    }
    private void updateEisodosList(int y1, int y2, int y0, int midLength) { //yparxei
        int i = 1;
        for (Eisodos eisodos : eisodosList) {
            if (eisodos.pos.y >= y1 && eisodos.pos.y < y2) {
                eisodos.pos = new Point(0, y0 + (i - 1)*(breadth + midLength));
                eisodos.breadth = breadth;
                i++;
            }
        }
    }

    //-----
    // getCount
    //-----
    public static byte getCount() { return count; }

    //-----
    // getBreadth
    //-----

```



```

public int getBreadth() { return breadth; }
}

```

Γ.22 Αρχείο Exodos.java

```

//*****
// Java Class Name : Exodos
//*****
package grpekast.sch.users.model;

public class Exodos extends Stoixeio {
    public boolean epityxis = false;
    public final transient int left, top;

    //-----
    // Constructor
    //-----
    public Exodos(int x, int y, int width, int height) {
        super(x, y, width, height);
        left = (int)rect.left;
        top = (int)rect.top;
    }
}

```

Γ.23 Αρχείο Box.java

```

//*****
// Java Class Name : Box
//*****
package grpekast.sch.users.model;

public class Box {
    public enum Bonus {TIPOTA, SFIGGA, SFIGGES, SFIGGA9, ZARI}
    public final Bonus perioxomeno;
    public final boolean dikaioima;
    public final byte No; //box number
    private static final String[] strTitleInfo = new String[] {
        "Καμία ερώτηση.",
        "1 ερώτηση Πολλαπλής-Επιλογής με δικαίωμα επιλογής κατηγορίας.",
        "2 ερωτήσεις Πολλαπλής-Επιλογής.",
        "1 ερώτηση Πίνακα πολλαπλών σωστών απαντήσεων.",
        "Ζάρι - Τυχαίο ρίξιμο."};
    private static final String[] strBoxInfo = new String[] {
        "Συνέχεια στό λαβύρινθο χωρίς πρόκληση μόνους ζωής..",
        "Αν απαντήσεις...\n- ΣΩΣΤΑ: Κερδίζεις 1 ζωή.\n- ΛΑΘΟΣ: Χάνεις 1 ζωή.",
        "Αν απαντήσεις σωστά...\n- 0 ΦΟΡΕΣ: Χάνεις 1 ζωή.\n" +
        "- 1 ΦΟΡΑ: Δεν κερδίζεις/χάνεις τίποτα.\n- 2 ΦΟΡΕΣ: Κερδίζεις 2 ζωές.",
        "Αν απαντήσεις...\n- ΣΩΣΤΑ: Κερδίζεις 2 ζωές.\n- ΛΑΘΟΣ: Χάνεις 1 ζωή.",
        "Με αποτέλεσμα ζαριού...\n- 1 ή 2: χάνεις 1 ζωή.\n" +
        "- 3 ή 4: Δεν κερδίζεις/χάνεις τίποτα.\n" +
        "- 5: Κερδίζεις 1 ζωή.\n- 6: Κερδίζεις 2 ζωές."};
    private static byte count = 0;

    //-----
    // Constructor
    //-----
    public Box() {
        if (count >= 5) count = 0;
        count++;
        No = count;
    }
}

```

```

        dikaioma = (No == 2);
        periexomeno = periexei();
    }

    //-----
    // periexei
    //-----
    private Bonus periexei() {
        Bonus content;
        switch (No) {
            case 2: content = Bonus.SFIGGA;
                break;
            case 3: content = Bonus.SFIGGES;
                break;
            case 4: content = Bonus.SFIGGA9;
                break;
            case 5: content = Bonus.ZARI;
                break;
            default:
                content = Bonus.TIPOTA;
        }
        return content;
    }

    //-----
    // getTitleInfo
    //-----
    public String getTitleInfo() { return strTitleInfo[No-1]; }

    //-----
    // getBoxInfo
    //-----
    public String getBoxInfo() { return strBoxInfo[No-1]; }
}

```

Γ.24 Αρχείο Score.java

```

//*****
// Java Class Name : Score
//*****
package grpekast.sch.users.model;

import android.support.annotation.NonNull;

public class Score {
    private static byte prospatheia = 3;
    private static int points = 0;

    //-----
    // katagrafetai
    //-----
    public static void katagrafetai(int pts) { points += pts; }

    //-----
    // anagrafetai
    @NonNull //-----
    public static String anagrafetai() { return String.valueOf(points); }

    //-----
    // athroizetaiProspath
    //-----
    public static void athroizetaiProspath() { prospatheia++; }

    //-----
    // xaneiProspath

```

```

//-----
public static byte xaneiProspath() { return --prospatheia; }

//-----
// getScore
//-----
public static int getScore() { return points; }

//-----
// getProspatheia
//-----
public static byte getProspatheia() { return prospatheia; }

//-----
// reset
//-----
public static void reset() {
    prospatheia = 3;
    points = 0;
}

public static void anakalyptetaiProspath() {}
}

```

Γ.25 Αρχείο State.java

```

//*****
// Java Abstract Class Name : State
//*****
package grpekast.sch.users.state;

import android.view.MotionEvent;

import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.network.Bluetooth;
import grpekast.sch.users.utility.Grafika;

public abstract class State {
    public boolean isThreaded = false;
    protected String mIntroFile="";
    protected static Bluetooth mBluetooth = null;

    //-----
    // setCurrentState
    //-----
    public void setCurrentState(State newState) {
        Platform.sGame.setCurrentState(newState); }

    //-----
    // setBackground
    //-----
    public abstract void setBackground(Grafika grafika);

    //-----
    // update
    //-----
    public abstract void update(float dt); // { }

    //-----
    // render
    //-----
    public abstract void render(Grafika grafika);

    //-----
    // onTouch

```

```

//-----
public abstract boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY);

//-----
// onSensor
//-----
public void onSensor(float accelX, float accelY) {}

//-----
// resumeThread
//-----
protected void resumeThread() { Platform.sGame.resumeThread(); }

//-----
// pauseThread
//-----
protected void pauseThread() { Platform.sGame.pauseThread(); }

//-----
// getIntroFile
//-----
public String fetchIntroFile() {
    String strIntroFile = mIntroFile;
    mIntroFile = "";
    return strIntroFile;
}
}

```

Γ.26 Αρχείο Avatar.java

```

//*****
// Java Class Name : Avatar
//*****
package grpekast.sch.users.state;

import android.graphics.Color;
import android.view.MotionEvent;

import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Score;
import grpekast.sch.users.network.Bluetooth;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.UIButton;

public class Avatar extends State {
    private UIButton[] btnAvatar;

    //-----
    // Constructor
    //-----
    public Avatar(Bluetooth bt) {
        int x, y, x1, dx, width, i = 0;
        mBluetooth = bt;
        Assets.load();
        Score.reset();
        btnAvatar = new UIButton[Assets.beastie.length];

        //----- place on 6 columns -----
        x1 = (Platform.GAME_WIDTH - 6 * Assets.beastie[0].getWidth()) / 7;
        dx = x1 + Assets.beastie[0].getWidth();
        width = Platform.GAME_WIDTH - Assets.beastie[0].getWidth() - x1;

        // Initialize the buttons at the proper coordinates (6 columns).
        for (y = 45; i < btnAvatar.length; y += dx) {

```

```

        for (x = x1; x < width; x += dx) {
            btnAvatar[i] = new UIButton(x, y, x + 20, y + 20, Assets.beastie[i]);
            i++;
            if (i >= btnAvatar.length) break;
        }
    }

    @Override
    public void setBackground(Grafika grafika) {}

    @Override
    public void update(float dt) {}

    @Override
    public void render(Grafika grafika) {
        renderBlackboard(grafika);
        renderTitle(grafika);
        for (UIButton btn : btnAvatar) btn.render(grafika);
    }

    @Override
    public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
        switch (myEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                for (UIButton btn : btnAvatar) btn.onTouchDown(scaledX, scaledY);
                break; // Only one button should be active
            case MotionEvent.ACTION_UP:
                for (int i = 0; i < btnAvatar.length; i++) {
                    if (btnAvatar[i].isPressed(scaledX, scaledY)) {
                        btnAvatar[i].cancel();
                        setCurrentState(new Labyrinth(i));
                        //setCurrentState(new BonusFasi(new Player(0, 0, 20, 20), i));
                        break;
                    }
                    else { //If the finger was released anywhere else...
                        btnAvatar[i].cancel(); //...cancel all actions.
                    }
                }
            }
        }
        return true;
    }

    private void renderBlackboard(Grafika grafika) {
        grafika.setColor(Color.rgb(0, 0, 0));
        grafika.fillRect(0, 0, Platform.GAME_WIDTH, Platform.GAME_HEIGHT);
    }

    private void renderTitle(Grafika grafika) {
        grafika.setTextSize(16);
        grafika.setColor(Color.rgb(200, 200, 175));
        int x = (Platform.GAME_WIDTH-grafika.getStringWidth("ΔΙΑΛΕΞΕ ΠΑΙΚΤΗ"))/2;
        grafika.drawString("ΔΙΑΛΕΞΕ ΠΑΙΚΤΗ", x, 20);
    }
}

```

Γ.27 Αρχείο Labyrinth.java

```

//*****
// Java Class Name : Labyrinth
//*****
package grpekast.sch.users.state;

import android.graphics.Color;
import android.graphics.Path;
import android.graphics.Point;

```

```

import android.graphics.RectF;
import android.graphics.Region;
import android.util.Log;
import android.view.MotionEvent;

import java.util.ArrayList;
import java.util.Random;

import static grpekast.sch.users.state.Labyrinth.Status.*;
import static grpekast.sch.users.model.Player.Fasi;

import grpekast.sch.users.database.Basi;
import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.labyrinthos.SpatialHashGrid;
import grpekast.sch.users.network.Syndesi;
import grpekast.sch.users.model.*;
import grpekast.sch.users.model.Diadromi.Way;
import grpekast.sch.users.utility.Grafika;

public class Labyrinth extends State {
    protected enum Status {NONE, START, MOVE, SCROLLX, SCROLLY, SFIGGA, KARDOYLA, KILL, END}
    private Status status = START;
    public final Point currentClip = new Point(0, 0);
    private static byte epomenos = 2;
    private static final byte DIADROMI_COUNT = 7;
    private static final byte BREADTHWAY = 28;
    private static final int PLAYER_WIDTH = 20;
    private static final int PLAYER_HEIGHT = 20;
    private static final int SFIGGA_WIDTH = 24;
    private static final int SFIGGA_HEIGHT = 24;
    private static final int KARDOYLA_LENGTH = 16;
    private static Random rand = new Random();
    private final int LABYRINTH_TOP, LABYRINTH_WIDTH, LABYRINTH_HEIGHT;
    private int dx = 0, dy = 0;
    private Player player;
    private Sfigga sfigga;
    private Stoixeio stoixeio, kardoula;
    private ArrayList<Sfigga> sminos;
    private ArrayList<Stoixeio> kardoules;
    private SpatialHashGrid gridSfigges, gridkardoules;
    private Eisodos[] eisodos;
    private Exodos exodos;
    private Diadromi[] diadromi;
    private int scrollX, scrollY, nextHeartX, nextHeartY;
    private int exitId;
    private Path path;
    private Region pathway, playerIntersect;
    private double theta = -225; // μοίρες
    private byte dyskolia;

    //-----
    // Constructors
    //-----
    public Labyrinth(int playerImageIndex) { this(null, playerImageIndex); }
    public Labyrinth(Player player, int playerImageIndex) {
        diadromi = new Diadromi[DIADROMI_COUNT];
        eisodos = new Eisodos[DIADROMI_COUNT];
        path = new Path();
        pathway = new Region(); playerIntersect = new Region();
        sminos = new ArrayList<>(); kardoules = new ArrayList<>();
        dyskolia = (byte) (epomenos%2);
        createLabyrinth();

        int top = pathway.getBounds().top;
        int bottom = pathway.getBounds().bottom;
        LABYRINTH_TOP = top - 2 * Platform.GAME_HEIGHT / 3;
        LABYRINTH_WIDTH = 2 * Platform.GAME_WIDTH;
        LABYRINTH_HEIGHT = -LABYRINTH_TOP + bottom + (Platform.GAME_HEIGHT

```

```

        - (bottom - Platform.GAME_HEIGHT)% Platform.GAME_HEIGHT);

    gridSfigges = new SpatialHashGrid(LABYRINTH_TOP
        , LABYRINTH_WIDTH, LABYRINTH_HEIGHT, 4*SFIGGA_WIDTH);
    gridkardoules = new SpatialHashGrid(LABYRINTH_TOP
        , LABYRINTH_WIDTH, LABYRINTH_HEIGHT, 4 * SFIGGA_WIDTH);
    for (Sfigga sfigga: sminos) gridSfigges.insertStaticObject(sfigga);
    for (Stoixeio kardoula: kardoules) gridkardoules.insertStaticObject(kardoula);

    exitId = 2*(epomenos%2); // Δείκτης εικόνας κλειστού σεντουκιού/πόρτας

    createPlayer(player, playerImageIndex);
    if (epomenos == 2) mIntroFile = "ppanther_cut1.mp4";
    else status = MOVE;
}

//-----
// setBackground
@Override //-----
public void setBackground(Grafika grafika) {
    if (status == START) {
        showInfoScreen(grafika);
        return;
    }
    if (status == SFIGGA) {
        grafika.clear(Color.rgb(0, 200, 0));
        if (grafika.getSaveCount() > 1) grafika.restore();
        grafika.setColorFilter(null);
        grafika.setColor(Color.rgb(255, 255, 63));
        update(0f);
    }
    grafika.setFontDefault(12);
    renderGrass(grafika);
    renderPath(grafika);
    renderSfigga(grafika);
    renderHeart(grafika);
    renderScore(grafika);
}

//-----
// update
@Override //-----
public void update(float dt) {
    switch (status) {
        case MOVE:
            if (playerStepsOnEdge()) {
                player.isMoving = false;
                scrollScreen();
                break;
            }
            player.move(dt);
            if (playerStepsOnGrass()) {
                player.isMoving = (player.pushBack() < 3);
                //Assets.playSound(Assets.crushID);
            }
            else if (playerStepsOnSfigga()) {
                player.pefteiSeSfigga();
                Assets.playSound(Assets.sfiggaID);
                currentClip.set(dx, dy);
                sfigga.kaneiErotisi(this, player);
                status = SFIGGA;
            }
            else if (playerStepsOnKardoula()) {
                kardoula.rect.offset((nextHeartX-kardoula.rect.left) % 16
                    , (nextHeartY-kardoula.rect.top) % 16);
                scrollX = (int) (nextHeartX-kardoula.rect.left) / 16;
                scrollY = (int) (nextHeartY-kardoula.rect.top) / 16;
                Assets.playSound(Assets.lifeID);
                status = KARDOYLA;
            }
    }
}

```

```

else if (playerStepsOnExit()) {
    if (player.synexizei()) {
        if (epomenos == 2) setCurrentState(new BonusFasi(player));
        else setCurrentState(new GameOver(player.avatarId));
        epomenos = (byte)(epomenos%2 + 1);
    }
}
if (!player.isMoving) pauseThread();
break;

case SCROLLX:
    if (Math.abs(dx - currentClip.x) <= 2 * Platform.GAME_WIDTH / 3) {
        dx += scrollX;
        break; }
    status = MOVE;
    pauseThread();
    break;
case SCROLLY:
    if (Math.abs(dy - currentClip.y) <= 2 * Platform.GAME_HEIGHT / 3) {
        dy += scrollY;
        break; }
    status = MOVE;
    pauseThread();
    break;
case KARDOYLA:
    if (kardoula.rect.top > nextHeartY) {
        kardoula.rect.offset(scrollX, scrollY);
        break; }
    gridkardoules.removeObject(kardoula);
    kardoules.remove(kardoula);
    Score.athroizetaiProspath();
    status = NONE;
    break;
case KILL:
    if (theta < 136) {
        theta += 5;
        break; }
    gridSfigges.removeObject(sfigga);
    sminos.remove(sfigga);
    theta = -225;
    isThreaded = false;
    status = MOVE;
    pauseThread();
    break;
case SFIGGA:
    int result;
    if ((result = sfigga.getApotelesma()) > 0) {
        Score.katagrafetai(result);
        if (player.synexizei()){
            exitId = 1 + 2*(epomenos%2); //εικόνα ανοικτού σεντιουκιού/πόρτας
            Assets.playSound(Assets.latchID); }
        status = KILL;
        stoixeio = sfigga;
        isThreaded = true;
        break; }
    if (Score.xaneiProspath() < 0) {
        Assets.playSound(Assets.lifeEmptyID);
        status = END;
        stoixeio = player;
        scrollX = PLAYER_WIDTH*8; scrollY = PLAYER_HEIGHT*8;
        isThreaded = true;
        break; }
    status = NONE;
    break;
case END:
    if (scrollX <= 0)
        setCurrentState(new GameOver(player.avatarId));
    scrollX--; scrollY--;
case START:
    break;

```



```

        default:
            status = MOVE;
        }
    }

    //-----
    // render
    @Override //-----
    public void render(Grafika grafika) {
        switch (status) {
            case MOVE:
                break;
            case SCROLLX:
            case SCROLLY:
                grafika.scrollScreen(scrollX, scrollY);
            case KARDOYLA:
                renderGrass(grafika);
                renderPath(grafika);
                renderSfigga(grafika);
                renderHeart(grafika);
                renderScore(grafika);
                break;
            case KILL:
                renderKill(stoixeio, grafika);
                renderHeart(grafika);
                break;
            case END:
                renderPlayerEnd(grafika);
            case START:
                return;
            default:
                setBackground(grafika);
        }
        renderPlayer(grafika);
    }

    //-----
    // onTouch
    @Override //-----
    public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
        if (status == START) { status = MOVE;
            setCurrentState(this);
            Assets.playSound(Assets.startID);
            return false;
        }
        if (status != MOVE) return false;
        scaledX += -dx; scaledY += -dy;

        switch (myEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                if (!pathway.contains(scaledX, scaledY)) return false;
                break;
            case MotionEvent.ACTION_UP:
                player.topotheteitai(scaledX, scaledY);
                resumeThread();
        }
        return true;
    }

    //-----
    // onSensor
    @Override //-----
    public void onSensor(float ax, float ay) {
        if (player.isMoving || status != MOVE) return;

        if (ax > 10) {
            player.topotheteitai(player.getX(), LABYRINTH_TOP+LABYRINTH_HEIGHT);
            resumeThread();
        }
        else if (ax < -7) {

```

```

        player.topotheteitai(player.getX(), LABYRINTH_TOP);
        resumeThread();
    }
    else if (ay > 7) {
        player.topotheteitai(LABYRINTH_WIDTH, player.getY());
        resumeThread();
    }
    else if (ay < -7) {
        player.topotheteitai(1, player.getY());
        resumeThread();
    }
}

//-----
// createLabyrinth
//-----
private void createLabyrinth() { // dimiourgeitai
    int i;
    Eisodos.init();
    for (i = 0; i < DIADROMI_COUNT; i++) {
        eisodos[i] = new Eisodos(-160, Platform.GAME_HEIGHT+160, BREADTHWAY);

        if (mBluetooth != null) {
            if (Syndesi.isServer()) {
                diadromi[i]=new Diadromi((3*Platform.GAME_WIDTH/2)/BREADTHWAY);
                Log.d("Diadromi"+i+" MHKOS: ", ""+diadromi[i].MHKOS);
                mBluetooth.write(diadromi[i]);
            } else diadromi[i] = (Diadromi) Platform.sObject.remove(0);
        }
        else
            diadromi[i] = new Diadromi((3*Platform.GAME_WIDTH/2) / BREADTHWAY);
    }
    createPath();
}

//-----
// createPath
//-----
private void createPath() {
    Way[][] ways = new Way[DIADROMI_COUNT][];
    int exitNo = rand.nextInt(DIADROMI_COUNT);
    int i, j, x, y, breadth;
    int top = 0, bottom = 0;
    int verticalPathX, sfiggaCount;

    path.setFillType(Path.FillType.WINDING);
    breadth = BREADTHWAY;
    verticalPathX = (int) ((Platform.GAME_WIDTH / 2) / breadth) * breadth;

    for (i = 0; i < DIADROMI_COUNT; i++) {
        ways[i] = diadromi[i].getPath();
        x = eisodos[i].pos.x; // (= 0)
        y = eisodos[i].pos.y;
        path.addRect(x, y, x + breadth, y + breadth, Path.Direction.CW);
        sfiggaCount = 0;
        for (j = 0; j < diadromi[i].MHKOS; j++) {
            switch (ways[i][j]) {
                case DOWN:
                    y = y + breadth;
                    break;
                case UP:
                    y = y - breadth;
                    break;
                default:
                    x = x + breadth;
            } //---Propable Sfigga addition after the vertical line-----
            if (x > verticalPathX && (Syndesi.isServer() || mBluetooth == null)) {
                if (x <= verticalPathX + breadth && sfiggaCount == 0)
                    sfiggaCount += createSfigga(
                        x + breadth/2, y + breadth/2, (byte)rand.nextInt(5));
            }
        }
    }
}

```

```

        else if ((j < diadromi[i].MHKOS - 1)
            && (rand.nextInt(diadromi[i].MHKOS - 1) < 3))
            sfiggaCount += createSfigga(
                x + breadth/2, y + breadth/2, (byte)rand.nextInt(5));
//-----
        else if (rand.nextInt(diadromi[i].MHKOS)==3)//Propable kardoula
            createHeart(x + breadth / 2, y + breadth / 2);
//-----
    } //-----
    if (y < top) top = y;
    else if (y > bottom) bottom = y;
    path.addRect(x, y, x + breadth, y + breadth, Path.Direction.CW);
}
if (exitNo == i)
    exodos = new Exodos(
        x + breadth/ 2, y + breadth/2, BREADTHWAY-4, BREADTHWAY-4);
}

if (mBluetooth != null) {
    if (Syndesi.isServer()) {
        mBluetooth.write(exodos.left);
        mBluetooth.write(exodos.top);
    } else readStoixeia();
}
bottom += breadth;
path.addRect(verticalPathX, top, verticalPathX+breadth, bottom
    , Path.Direction.CW);
pathway.setPath(path,
    new Region(0, top, 3*Platform.GAME_WIDTH/2 + 2*breadth, bottom));
}

//-----
// createPlayer
//-----
private void createPlayer(Player p, int pIndex) {
    int r = rand.nextInt(DIADROMI_COUNT);
    int playerX = eisodos[r].pos.x + 4;
    int playerY = eisodos[r].pos.y + 4;

    if (p != null) {
        player = p;
        player.set(playerX, playerY);
    } else {
        player = new Player(playerX,playerY,PLAYER_WIDTH,PLAYER_HEIGHT,pIndex);
        epomenos = 2;
    }

    player.eiserxetai(Fasi.LABYRINTHOS);
}

//-----
// createSfigga
//-----
private int createSfigga(int x, int y, byte categ) {
    int gridX = (int)Math.floor((float)x / BREADTHWAY);
    int gridY = (int)Math.floor((float)y / BREADTHWAY);
    for (Sfigga sfg: sminos)
        if (sfg.grid.x == gridX && sfg.grid.y == gridY) return 0;
    Sfigga sfigga = new Sfigga(x, y, SFIGGA_WIDTH, SFIGGA_HEIGHT
        , categ, dyskolia);
    sfigga.grid.x = gridX;
    sfigga.grid.y = gridY;
    sminos.add(sfigga);
    if (mBluetooth != null) mBluetooth.write(sfigga);
    return 1;
}

//-----
// createHeart
//-----
private void createHeart(int x, int y) {

```

```

int gridX = (int)Math.floor((float)x / BREADTHWAY);
int gridY = (int)Math.floor((float)y / BREADTHWAY);
for (Stoixeio st: kardoules)
    if (st.grid.x == gridX && st.grid.y == gridY) return;
Stoixeio kardoula = new Stoixeio(x, y, KARDOYLA_LENGTH, KARDOYLA_LENGTH);
kardoula.grid.x = gridX;
kardoula.grid.y = gridY;
kardoules.add(kardoula);

if (mBluetooth != null && Syndesi.isServer()) {
    mBluetooth.write(kardoula);
    mBluetooth.write(String.valueOf(x).concat(",")
        .concat(String.valueOf(y)));
}

//-----
// readStoixeia
//-----
private void readStoixeia() {
    Stoixeio stoixeio;
    Sfigga sfigga;
    int x, y;

    while (!Platform.sObject.isEmpty() && Platform.sObject.size()>2) {
        try { stoixeio = (Stoixeio) Platform.sObject.remove(0); }
        catch (Exception e) { break; }

        if (stoixeio instanceof Sfigga) {
            sfigga = (Sfigga) stoixeio;
            x = sfigga.left + SFIGGA_WIDTH / 2;
            y = sfigga.top + SFIGGA_HEIGHT / 2;
            sfigga.reconstruct(x, y, SFIGGA_WIDTH, SFIGGA_HEIGHT);
            sfigga.grid.x = (int) Math.floor((float) x / BREADTHWAY);
            sfigga.grid.y = (int) Math.floor((float) y / BREADTHWAY);
            sminos.add(sfigga);
        }
        else { String strPos = (String) Platform.sObject.remove(0);
            String[] coord = strPos.split(",");
            x = Integer.parseInt(coord[0]);
            y = Integer.parseInt(coord[1]);
            createHeart(x, y);
        }
    }
    x = (int) Platform.sObject.remove(0);
    y = (int) Platform.sObject.remove(0);
    exodos = new Exodos(x-2 + BREADTHWAY/2, y-2 + BREADTHWAY/2
        , BREADTHWAY-4, BREADTHWAY-4);
}

//-----
// renderGrass
//-----
private void renderGrass(Grafika grafika) {
    grafika.setColor(Color.rgb(0, 200, 0));
    grafika.fillRect(0, LABYRINTH_TOP, LABYRINTH_WIDTH, LABYRINTH_HEIGHT);
}

//-----
// renderPath
//-----
private void renderPath(Grafika grafika) {
    grafika.setColor(Color.rgb(255, 255, 63)); //Κίτρινο
    grafika.drawImage(path);
}

//-----
// renderScore
//-----
private void renderScore(Grafika grafika) {
    String strScore = "\u03A3\u0391\u039F\u03A1 " + Score.anagrafetai();
}

```

```

int width, height = grafika.getStringHeight(strScore);
int x = Platform.GAME_WIDTH - dx - 8 - grafika.getStringWidth(strScore);
int y = height + 3 - dy;
byte prosopatheiaCount = Score.getProspatheia();

grafika.setColor(Color.BLACK);
grafika.drawString(strScore, x, y);

x = 5 - dx;
y = 1 - dy;
width = Assets.heart.getWidth();
for (byte i = 0; i < prosopatheiaCount; i++, x+=width)
    grafika.drawImage(Assets.heart, x, y);
nextHeartX = x;
nextHeartY = y;
grafika.setColor(Color.rgb(255, 255, 63)); //Κίτρινο
}

//-----
// renderPlayer
//-----
private void renderPlayer(Grafika grafika) {
    grafika.setColor(Color.rgb(255, 255, 63));
    grafika.fillRect(player.getOldX(), player.getOldY()
        , PLAYER_WIDTH, PLAYER_HEIGHT);
    renderDoor(grafika);
    if (player.getOldX() > player.getX())
        grafika.drawFlipImage(Assets.beastie[player.avatarId],
            player.getX(), player.getY(), PLAYER_WIDTH, PLAYER_HEIGHT);
    else
        grafika.drawImage(Assets.beastie[player.avatarId],
            player.getX(), player.getY(), PLAYER_WIDTH, PLAYER_HEIGHT);
}

//-----
// renderPlayerEnd
//-----
private void renderPlayerEnd(Grafika grafika) {
    int width = scrollX / 8;
    int height = scrollY / 8;
    int x = (int)player.position.x - width/2;
    int y = (int)player.position.y - height/2;
    grafika.setColor(Color.rgb(255, 255, 63));
    grafika.fillRect(player.getX(), player.getY(), PLAYER_WIDTH, PLAYER_HEIGHT);
    grafika.drawImage(Assets.beastie[player.avatarId], x, y, width, height);
}

//-----
// renderSfigga
//-----
private void renderSfigga(Grafika grafika) {
    for (Sfigga sfigga : sminos)
        grafika.drawImage(Assets.sphinx[sfigga.category],
            sfigga.left, sfigga.top, SFIGGA_WIDTH, SFIGGA_HEIGHT);
}

//-----
// renderHeart
//-----
private void renderHeart(Grafika grafika) {
    for (Stoixeio kardoula : kardoules)
        grafika.drawImage(Assets.heart
            , (int) kardoula.rect.left, (int) kardoula.rect.top
            , KARDOYLA_LENGTH, KARDOYLA_LENGTH);
}

//-----
// renderDoor
//-----
private void renderDoor(Grafika grafika) {

```

```

        grafika.drawImage(Assets.exit[exitId], exodos.left, exodos.top
            , BREADTHWAY-4, BREADTHWAY-4);
    }

    //-----
    // renderKill
    //-----
    private void renderKill(Stoixeio stoixeio, Grafika grafika) {
        float x1 = stoixeio.position.x
            + (BREADTHWAY/2f)*(float)Math.cos(theta*Math.PI/180);
        float y1 = stoixeio.position.y
            + (BREADTHWAY/2f)*(float)Math.sin(theta*Math.PI/180);
        grafika.drawLine(x1, y1, stoixeio.position.x, stoixeio.position.y);
        if (theta > 134d) grafika.fillRect((int) stoixeio.position.x-BREADTHWAY/2
            , (int)stoixeio.position.y-BREADTHWAY/2
            , BREADTHWAY, BREADTHWAY);
    }

    //-----
    // scrollScreen
    //-----
    private void scrollScreen() {
        if (player.getX() != player.getOldX()) {
            currentClip.x = dx;
            scrollX = player.getOldX() > player.getX() ? 8 : -8; scrollY = 0;
            status = Status.SCROLLX;
        } else {
            currentClip.y = dy;
            scrollX = 0; scrollY = player.getOldY() > player.getY() ? 8 : -8;
            status = Status.SCROLLY;
        }
        resumeThread();
        update(0f);
    }

    //-----
    // playerStepsOnEdge
    //-----
    private boolean playerStepsOnEdge() {
        return ((player.getY()+dy <= PLAYER_HEIGHT/2)
            || (player.getY()+dy >= Platform.GAME_HEIGHT-PLAYER_HEIGHT-PLAYER_HEIGHT/2)
            || ((player.getX()+dx <= 0) && (player.getX() > 0))
            || (player.getX()+dx >= Platform.GAME_WIDTH - PLAYER_WIDTH));
    }

    //-----
    // playerStepsOnGrass
    //-----
    private boolean playerStepsOnGrass() {
        return playerIntersect.op(player.getRect(), pathway, Region.Op.DIFFERENCE);
    }

    //-----
    // playerStepsOnSfigga
    //-----
    private boolean playerStepsOnSfigga() {
        ArrayList<Stoixeio> colliders = gridSfigges.getPotentialColliders(player);
        for (Stoixeio collider : colliders) {
            if (RectF.intersects(player.rect, collider.rect)) {
                sfigga = (Sfigga)collider;
                return true;
            }
        }
        return false;
    }

    //-----
    // playerStepsOnKardoula
    //-----
    private boolean playerStepsOnKardoula() {
        ArrayList<Stoixeio> colliders=gridkardoules.getPotentialColliders(player);
    }

```

```

        for (Stoixeio collider : colliders) {
            if (RectF.intersects(player.rect, collider.rect)) {
                kardoula = collider;
                return true;
            }
        } return false;
    }

    //-----
    // playerStepsOnExit
    //-----
    private boolean playerStepsOnExit() {
        return player.rect.contains(exodos.position.x, exodos.position.y);
    }

    //-----
    // showInfoScreen
    //-----
    private void showInfoScreen(Grafika grafika) {
        grafika.clear(Color.BLACK);

        grafika.setTextSize(14);
        grafika.setColor(Color.rgb(200, 200, 175));
        int x=(Platform.GAME_WIDTH-grafika.getStringWidth("ΚΑΤΗΓΟΡΙΕΣ ΕΡΩΤΗΣΕΩΝ"))/2;
        grafika.drawString("ΚΑΤΗΓΟΡΙΕΣ ΕΡΩΤΗΣΕΩΝ", x, 20);

        grafika.setFontDefault(12);
        for (int i = 0, y = 25; i< 5; i++, y+=38) {
            grafika.drawImage(Assets.sphinx[i], 20, y, 32, 32);
            grafika.drawString("==> " + Basi.getCategories(i), 60, y + 20);
        }
        grafika.drawString("Πατήστε στην οθόνη για συνέχεια."
            , 10, Platform.GAME_HEIGHT - 8);
    }

    public byte getLabyrNo() { return (byte)(epomenos%2 + 1); }
}

```

Γ.28 Αρχείο Eidos.java

```

//*****
// Java Abstract Class Name : Eidos
//*****
package grpekast.sch.users.state;

import android.graphics.Color;
import android.graphics.ColorFilter;
import android.graphics.Point;

import static grpekast.sch.users.utility.Emfanisi.*;

import grpekast.sch.users.database.Enigma;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Sfigga;
import grpekast.sch.users.utility.Emfanisi;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.UIButton;

public abstract class Eidos extends State {
    protected enum Status { SCROLL, DETAIL, SELECT, MESSAGE, SUCCESS, FAIL, EXIT }
    protected Status status;
    protected State sourceState;
    protected Sfigga sfigga;
    protected String strCategory, strQuestion;
    protected UIButton[] btnAnswer;
}

```

```

protected boolean[] isCorrect;
protected UIButton btnContinue = new UIButton(240, 220, 68
    , "ΕΥΝΕΧΕΙΑ...", 12
    , Color.BLACK, Color.rgb(200, 200, 175), Color.rgb(102, 102, 71));
protected Point currentClip;
protected int dx, dy, scrollX, scrolly;
protected final int SCROLLSTEPS = 16;
protected String message, strNotification = "";

protected ColorFilter mColorFilter;
protected String strEidos;

//-----
// init (abstract)
//-----
protected abstract void init();

//-----
// setCategory
//-----
public void setCategory(byte category) {
    switch (category) {
        case 1: mColorFilter = Emfanisi.adjustHue(COLORFILTER_1); break;
        case 2: mColorFilter = Emfanisi.adjustHue(COLORFILTER_2); break;
        case 3: mColorFilter = Emfanisi.adjustHue(COLORFILTER_3); break;
        case 4: mColorFilter = Emfanisi.adjustHue(COLORFILTER_4);
    }
}

//-----
// provalei (abstract)
//-----
public abstract void provalei(Sfigga sfigga, Enigma enigma);

//-----
// renderBackground (abstract)
//-----
protected abstract void renderBackground(Grafika grafika);

//-----
// renderQuestion (abstract)
//-----
protected abstract void renderQuestion(Grafika grafika);

//-----
// renderAnswers (abstract)
//-----
protected abstract void renderAnswers(Grafika grafika);

//-----
// renderMessage
//-----
protected void renderMessage(Grafika grafika) {
    grafika.setFontDefault(12);
    int wdh = grafika.getStringWidth(message);
    int hght = grafika.getStringHeight(message);
    int centerX = (Platform.GAME_WIDTH - wdh) / 2;
    int centerY = (Platform.GAME_HEIGHT / 2) - 5;
    grafika.setColor(Color.GRAY);
    grafika.fillRect(centerX - 2, centerY - hght - 1, wdh + 4, hght + 3);
    grafika.setColor(Color.BLACK);
    grafika.drawString(message, centerX, centerY);
}

//-----
// renderNotification
//-----
protected void renderNotification(Grafika grafika) {
    int wdh = grafika.getStringWidth(strNotification);
    int rightX = Platform.GAME_WIDTH - wdh - 10;
}

```



```

        int centerY = (Platform.GAME_HEIGHT / 2) - 5;
        grafika.setColor(Color.BLACK);
        grafika.drawString(strNotification, rightX, centerY);
    }

    //-----
    // renderContinue
    //-----
    protected void renderContinue(Grafika grafika) { btnContinue.render(grafika); }

    //-----
    // setNotification
    //-----
    public void setNotification(String str) { strNotification = str; }
}

```

Γ.29 Αρχείο MultipleChoice.java

```

//*****
// Java Class Name : MultipleChoice
//*****
package grpekast.sch.users.state;

import static grpekast.sch.users.state.Eidos.Status.*;

import android.graphics.Color;
import android.graphics.Point;
import android.view.MotionEvent;

import java.util.ArrayList;
import java.util.Collections;

import grpekast.sch.users.database.Apantisi;
import grpekast.sch.users.database.Enigma;
import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Sfigga;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.UIButton;

public class MultipleChoice extends Eidos {

    //-----
    // Constructors
    //-----
    public MultipleChoice(State state) {
        sourceState = state;
        currentClip = new Point(0, 0);
        init();
        isThreaded = false;
        status = DETAIL;
    }
    public MultipleChoice(State state, Point currentClip) {
        sourceState = state;
        this.currentClip = currentClip;
        init();
        isThreaded = true;
        status = SCROLL;
    }

    //-----
    // init
    //-----
    @Override //-----
    protected void init() {

```

```

        btnAnswer = new UIButton[4];
        isCorrect = new boolean[4];
        strEidos = "Multiple-Choice";

        dx = currentClip.x - currentClip.x%SCROLLSTEPS;
        dy = currentClip.y - currentClip.y%SCROLLSTEPS;
        scrollX = -dx / SCROLLSTEPS;
        scrollY = -dy / SCROLLSTEPS;
    }

    //-----
    // provalei
    @Override //-----
    public void provalei(Sfigga sfigga, Enigma enigma) {
        this.sfigga = sfigga;
        strCategory = enigma.erotisi.getCategory();
        strQuestion = enigma.erotisi.string;
        ArrayList<Apantisi> apantiseis = new ArrayList<>(4);
        for (int i = 0; i < 4; i++) apantiseis.add(enigma.apantisi[i]);

        Collections.shuffle(apantiseis);

        for (int i = 0, x = 4; i < 4; i++, x+=79) {
            btnAnswer[i] = new UIButton(x, 160, 75, apantiseis.get(i).string, 11
                , Color.BLACK, Color.rgb(255, 255, 63)
                , (isCorrect[i] = apantiseis.get(i).sosto)
                ? Color.BLUE : Color.RED);
        }
        setCurrentState(this);
    }

    //-----
    // setBackground
    @Override //-----
    public void setBackground(Grafika grafika) {
        grafika.save();
        grafika.scrollScreen(-currentClip.x%SCROLLSTEPS, -currentClip.y%SCROLLSTEPS);
        grafika.setFont(Assets.comicsans, 14);
        grafika.createTextbox(strQuestion, 200);
    }

    //-----
    // update
    @Override //-----
    public void update(float dt) {
        switch (status) {
            case SCROLL:
                if (dx != -scrollX || dy != -scrollY) {
                    dx += scrollX;
                    dy += scrollY;
                } else {
                    isThreaded = false;
                    status = DETAIL;
                    pauseThread();
                }
                break;
        }
    }

    //-----
    // render
    @Override //-----
    public void render(Grafika grafika) {
        switch (status) {
            case SCROLL:
                grafika.scrollScreen(scrollX, scrollY);
            case DETAIL:
                renderBackground(grafika);
                renderQuestion(grafika);
                if (status == SCROLL) break;
        }
    }

```

```

        renderAnswers(grafika);
        break;
    case SUCCESS:
    case FAIL:
        renderMessage(grafika);
        renderNotification(grafika);
        status = EXIT;
    default:
        renderContinue(grafika);
    }
}

//-----
// onTouch
@Override //-----
public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
    if (status == SCROLL) return false;

    if (myEvent.getAction() == MotionEvent.ACTION_DOWN) {
        for (int i = 0; i < 4; i++)
            btnAnswer[i].onTouchDown(scaledX, scaledY);
        btnContinue.onTouchDown(scaledX, scaledY);
    } // Only one button should be active
    if (myEvent.getAction() == MotionEvent.ACTION_UP) {
        if (btnContinue.isPressed(scaledX, scaledY))
            setCurrentState(sourceState);
        else if (status == EXIT);
        else if (!answerIsPressed(scaledX, scaledY))
            //the finger was released somewhere else - cancel all actions.
            for (int i = 0; i < 4; i++) btnAnswer[i].cancel();
    }
    return true;
}

//-----
// renderBackground
@Override //-----
protected void renderBackground(Grafika grafika) {
    int grassY = 20 + Assets.sphinxMedium.getHeight();
    grafika.setColor(Color.rgb(0, 123, 255));
    grafika.fillRect(0, 0, Platform.GAME_WIDTH, grassY);
    grafika.setColor(Color.rgb(0, 200, 0));
    grafika.fillRect(0, grassY, Platform.GAME_WIDTH, Platform.GAME_HEIGHT - grassY);
    grafika.drawImage(Assets.cloud, 1, 3);
    grafika.drawImage(Assets.cloud
        , Platform.GAME_WIDTH - Assets.cloud.getWidth()-30, 10);
    grafika.setColorFilter(mColorFilter);
    grafika.drawImage(Assets.sphinxMedium, 5, 20);
    grafika.setColorFilter(null);
}

//-----
// renderQuestion
@Override //-----
protected void renderQuestion(Grafika grafika) {
    //grafika.setFont(Assets.comicsans, 14);
    grafika.setColor(Color.BLACK);
    grafika.drawString(strCategory, 110, 30);
    grafika.drawTextbox(110, 45);
}

//-----
// renderAnswers
@Override //-----
protected void renderAnswers(Grafika grafika) {
    for (int i = 0; i < 4; i++) btnAnswer[i].render(grafika);
}

private boolean answerIsPressed(int x, int y) {
    for (int i = 0; i < 4; i++) {

```

```

        if (btnAnswer[i].isPressed(x, y)) {
            if (isCorrect[i]) {
                status = SUCCESS;
                Assets.playSound(Assets.successID);
                sfigga.lysiEnigmatos(1);
                message = "\u03A3\u03A9\u03A3\u03A4\u0399!";
            }
            else {
                status = FAIL;
                Assets.playSound(Assets.wrongID);
                sfigga.lysiEnigmatos(2);
                message = "\u039B\u0399\u0398\u039F\u03A3!";
            }
            return true;
        }
    } return false;
}
}
}

```

Γ.30 Αρχείο Pinax.java

```

//*****
// Java Class Name : Pinax
//*****
package grpekast.sch.users.state;

import static grpekast.sch.users.state.Eidos.Status.*;

import android.graphics.Color;
import android.graphics.Point;
import android.view.MotionEvent;

import java.util.ArrayList;
import java.util.Collections;

import grpekast.sch.users.database.Apantisi;
import grpekast.sch.users.database.Enigma;
import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Sfigga;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.UIButton;

public class Pinax extends Eidos {
    private long beforeUpdateRender;
    private int correctCount = 0, incorrectCount = 0;
    private int correctOldCount = 0, incorrectOldCount = 0;

    //-----
    // Constructors
    //-----
    public Pinax(State state) {
        sourceState = state;
        currentClip = new Point(0, 0);
        init();
        isThreaded = false;
        status = DETAIL;
    }
    public Pinax(State state, Point currentClip) {
        sourceState = state;
        this.currentClip = currentClip;
        init();
        isThreaded = true;
        status = SCROLL;
    }
}

```

```

//-----
// init
@Override //-----
protected void init() {
    btnAnswer = new UIButton[9];
    isCorrect = new boolean[9];
    strEidos = "Pinax";

    dx = currentClip.x - currentClip.x%SCROLLSTEPS;
    dy = currentClip.y - currentClip.y%SCROLLSTEPS;
    scrollX = -dx / SCROLLSTEPS;
    scrollY = -dy / SCROLLSTEPS;
}

//-----
// provalei
@Override //-----
public void provalei(Sfigga sfigga, Enigma enigma) {
    this.sfigga = sfigga;
    strCategory = enigma.erotisi.getCategory();
    strQuestion = enigma.erotisi.string;
    ArrayList<Apantisi> apantiseis = new ArrayList<>(9);
    for (int i = 0; i < 9; i++) apantiseis.add(enigma.apantisi[i]);

    Collections.shuffle(apantiseis);

    for (int i=0, x=5, y=130; i < 9; i++, x=5+105*(i%3), y=130+30*(i/3))
        btnAnswer[i] = new UIButton(x, y, 100, apantiseis.get(i).string, 11
            , Color.BLACK, Color.rgb(255, 255, 63)
            , (isCorrect[i] = apantiseis.get(i).sosto)
            ? Color.BLUE : Color.RED);
    setCurrentState(this);
}

//-----
// setBackground
@Override //-----
public void setBackground(Grafika grafika) {
    grafika.save();
    grafika.scrollScreen(-currentClip.x%SCROLLSTEPS, -currentClip.y%SCROLLSTEPS);
    grafika.setFont(Assets.comicsans, 14);
    grafika.createTextbox(strQuestion, 200);
}

//-----
// update
@Override //-----
public void update(float dt) {
    switch (status) {
        case SCROLL:
            if (dx != -scrollX || dy != -scrollY) {
                dx += scrollX;
                dy += scrollY;
            } else {
                isThreaded = false;
                status = DETAIL;
                pauseThread();
            } break;
        case SELECT:
            beforeUpdateRender = System.nanoTime();
            break;
        case MESSAGE:
            long updateDuration = System.nanoTime() - beforeUpdateRender;
            if (updateDuration > 2500000000L) {
                status = DETAIL;
                pauseThread(); }
    }
}
}

```

```

//-----
// render
@Override //-----
public void render(Grafika grafika) {
    switch (status) {
        case SCROLL:
            grafika.scrollScreen(scrollX, scrollY);
        case DETAIL:
            renderBackground(grafika);
            renderQuestion(grafika);
            if (status == SCROLL) break;
            renderAnswers(grafika);
            renderInfo(grafika);
            break;
        case SELECT:
            renderMessage(grafika);
            resumeThread();
            status = MESSAGE;
        case MESSAGE:
            break;
        case SUCCESS:
        case FAIL:
            renderMessage(grafika);
            renderNotification(grafika);
            status = EXIT;
        default:
            renderContinue(grafika);
    }
}

//-----
// onTouch
@Override //-----
public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
    if (status == SCROLL) return false;
    else if (status == MESSAGE) { status = DETAIL; pauseThread(); }

    if (myEvent.getAction() == MotionEvent.ACTION_DOWN) {
        for (int i = 0; i < 9; i++) {
            if (btnAnswer[i].isSelected()) continue;
            btnAnswer[i].onSelect(scaledX, scaledY);
        }
        btnContinue.onTouchDown(scaledX, scaledY);
    }
    if (myEvent.getAction() == MotionEvent.ACTION_UP) {
        if (btnContinue.isPressed(scaledX, scaledY))
            setCurrentState(sourceState);
        else if (status == EXIT);
        else if (allCorrect()) { //βρέθηκαν όλες οι σωστές απαντήσεις.
            status = SUCCESS;
            Assets.playSound(Assets.successID);
            setMessage(1);
            sfigga.lysiEnigmatos(incorrectCount); }
        else if (failure()) { //αποτυχία εύρεσης αρκετών σωστών απαντήσεων.
            status = FAIL;
            Assets.playSound(Assets.wrongID);
            setMessage(0);
            sfigga.lysiEnigmatos(incorrectCount); }
        else if (newSelection()) {
            if (correctSelection()) { //επιλέχθηκε 1 σωστή απάντηση.
                Assets.playSound(Assets.correctID);
                setMessage(1);
            } else { //επιλέχθηκε 1 λάθος απάντηση.
                Assets.playSound(Assets.wrongID);
                setMessage(0);
            }
        }
        status = SELECT;
    }
}
return true;

```

```

}

//-----
// renderBackground
@Override //-----
protected void renderBackground(Grafika grafika) {
    int grassY = 20 + Assets.sphinxMedium.getHeight();
    grafika.setColor(Color.rgb(0, 123, 255));
    grafika.fillRect(0, 0, Platform.GAME_WIDTH, grassY);
    grafika.setColor(Color.rgb(0, 200, 0));
    grafika.fillRect(0, grassY, Platform.GAME_WIDTH, Platform.GAME_HEIGHT - grassY);
    grafika.drawImage(Assets.cloud, 1, 3);
    grafika.drawImage(Assets.cloud
        , Platform.GAME_WIDTH - Assets.cloud.getWidth() - 30, 10);
    grafika.setColorFilter(mColorFilter);
    grafika.drawImage(Assets.sphinxMedium, 5, 20);
    grafika.setColorFilter(null);
}

//-----
// renderQuestion
@Override //-----
protected void renderQuestion(Grafika grafika) {
    grafika.setFont(Assets.comicsans, 14);
    grafika.setColor(Color.BLACK);
    grafika.drawString(strCategory, 110, 30);
    grafika.drawTextbox(110, 45);
}

//-----
// renderAnswers
@Override //-----
protected void renderAnswers(Grafika grafika) {
    for (int i = 0; i < 9; i++) btnAnswer[i].render(grafika);
}

//-----
// renderInfo
//-----
private void renderInfo(Grafika grafika) {
    // Επέλεξε τις 4 σωστές απαντήσεις.
    grafika.setFontDefault(11);
    grafika.setColor(Color.BLACK);
    grafika.drawString("Επέλεξε τις 4 σωστές απαντήσεις.", 30, 230);
}

//-----
// allCorrect
//-----
private boolean allCorrect() {
    correctOldCount = correctCount; incorrectOldCount = incorrectCount;
    correctCount = 0; incorrectCount = 0;
    for (int i = 0; i < 9; i++)
        if (btnAnswer[i].isSelected()) {
            if (isCorrect[i]) correctCount++; else incorrectCount++;
        }
    return correctCount == 4;
}

//-----
// failure
//-----
private boolean failure() { return incorrectCount > 1; }

//-----
// newSelection
//-----
private boolean newSelection() {
    return correctCount + incorrectCount > correctOldCount + incorrectOldCount; }

```

```

//-----
// correctSelection
//-----
private boolean correctSelection() { return correctCount > correctOldCount; }

//-----
// setMessage
//-----
private void setMessage(int msgId) {
    switch (msgId) {
        case 0: // Μήνυμα: ΛΑΘΟΣ!
            message = "ΛΑΘΟΣ!";
            break;
        case 1: // Μήνυμα: ΣΩΣΤΑ!
            message = "ΣΩΣΤΑ!";
    }
}
}

```

Γ.31 Αρχείο BonusFasi.java

```

//*****
// Java Class Name : BonusFasi
//*****
package grpekast.sch.users.state;

import android.graphics.Bitmap;
import android.graphics.Color;
import android.view.MotionEvent;

import java.util.Random;

import static grpekast.sch.users.state.BonusFasi.Status.*;
import static grpekast.sch.users.model.Player.Fasi;

import grpekast.sch.users.database.Basi;
import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Box;
import grpekast.sch.users.model.Player;
import grpekast.sch.users.model.Score;
import grpekast.sch.users.model.Sfigga;
import grpekast.sch.users.utility.Grafika;
import grpekast.sch.users.utility.UIButton;

public class BonusFasi extends State {
    protected enum Status { START, NONE, SELECT, CATEGORY, ZARI, END }
    private Status status = START;
    private final Box[] box = new Box[5];
    private final UIButton[] btnBox = new UIButton[5];
    private final UIButton[] txtBoxInfo = new UIButton[5];
    private UIButton btnSelect;
    private int boxNo = 0; //box selection number
    private int titleLeft, titleTop;
    private Player player;
    private Sfigga sfigga;
    private Zari zari;
    private float t = 0;

    //-----
    // Constructor
    //-----
    public BonusFasi(Player p) {
        this.player = p;
        player.eiserxetai(Fasi.BONUS);
    }
}

```



```

int boxWidth = Assets.box.getWidth();
int boxHeight = Assets.box.getHeight();
int columnWidth = (Platform.GAME_WIDTH-8) / 5;
int y = (Platform.GAME_HEIGHT - boxHeight) / 2;

for (int i = 0, x = 4; i < 5; i++, x+=columnWidth) {
    box[i] = new Box();
    btnBox[i] = new UIButton(x, y, x + boxWidth, y + boxHeight
        , Assets.box, Assets.boxOpen);
    txtBoxInfo[i] = new UIButton(5, 20 + (Platform.GAME_HEIGHT+boxHeight)/2
        , Platform.GAME_WIDTH-72-10-5, box[i].getBoxInfo(), 11
        , Color.rgb(102, 40, 0), Color.alpha(0), Color.alpha(0));
}
mIntroFile = "ppanther_cut2.mp4";
}

//-----
// setBackground
@Override //-----
public void setBackground(Grafika grafika) {
    if (status != START) return;
    String str = "ΜΙΟΝΟΥΣ ΦΑΕΗ";
    grafika.reset();

    grafika.clear(Color.rgb(200, 200, 175));
    grafika.setFontDefault(20);
    grafika.setColor(Color.rgb(102, 40, 0)); // BROWN
    grafika.drawString(str, (Platform.GAME_WIDTH-grafika.getStringWidth(str))/2
        , (Platform.GAME_HEIGHT - grafika.getStringHeight(str)) / 2);

    grafika.setTextSize(12);
    titleLeft = (Platform.GAME_WIDTH
        - grafika.getStringWidth("ΠΕΡΙΕΧΟΜΕΝΟ 5ου ΚΟΥΤΙΟΥ")) / 2;
    titleTop = Platform.GAME_HEIGHT / 6;

    btnSelect = new UIButton(
        Platform.GAME_WIDTH - 72 - 10
        , Platform.GAME_HEIGHT - (Platform.GAME_HEIGHT)/5, 72
        , "ΕΠΙΛΟΓΗ ΚΑΙ ΕΥΝΕΧΕΙΑ", 11
        , Color.rgb(200, 200, 175), Color.rgb(102, 40, 0), Color.BLACK);

    Assets.playSound(Assets.bonusFasiID);
    isThreaded = true;
}

//-----
// update
@Override //-----
public void update(float dt) {
    switch (status) {
        case START:
            t += dt;
            if (t > 3) {
                isThreaded = false;
                pauseThread();
                t = 0;
            }
            break;
        case ZARI:
            break;
        case END:
            kataxorisiBonus();
    }
}

//-----
// render
@Override //-----
public void render(Grafika grafika) {

```

```

if (isThreaded) return;

grafika.setColor(Color.rgb(200, 200, 175));
grafika.fillRect(0, 0, Platform.GAME_WIDTH, Platform.GAME_HEIGHT);

switch (status) {
case CATEGORY:
    renderCategories(grafika);
    return;
case SELECT:
    renderInfo(grafika);
    renderBoxContent(grafika);
    break;
case ZARI:
    renderZariBonus(grafika);
case END: return;
default:
    renderPrompt(grafika);
}
renderBox(grafika);
}

//-----
// onTouch
@Override //-----
public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
    if (isThreaded) { t = 3; return false; }

    switch (myEvent.getAction()) {
case MotionEvent.ACTION_DOWN:
        if (status == ZARI) break;
        if (status == CATEGORY) { // Επιλογή κατηγορίας, εάν ζητήθηκε -----
            for (UIButton btnCateg : txtBoxInfo)
                btnCateg.onTouchDown(scaledX, scaledY);
            break;
        } //-----
        boolean wasSelected; // Επιλογή κουτιού -----
        for (int i = 0; i < 5; i++) {
            wasSelected = btnBox[i].isSelected();
            btnBox[i].onSelect(scaledX, scaledY);
            if (btnBox[i].isSelected() && !wasSelected) {
                boxNo = box[i].No;
                Assets.playSound(Assets.boxOpenID);
                status = SELECT; }
        } //-----
        btnSelect.onTouchDown(scaledX, scaledY);
        break;
case MotionEvent.ACTION_UP:
        if (status == ZARI) {
            setCurrentState(new Labyrinth(player, player.icon)); return false;
        }
        if (status == CATEGORY) { //Επιβεβαίωση επιλογής κατηγορίας, εάν ζητήθηκε
            for (byte i = 0; i < 5; i++)
                if (txtBoxInfo[i].isPressed(scaledX, scaledY)) {
                    txtBoxInfo[i].cancel();
                    dinetaiEnigma(i);
                    return true;
                } else txtBoxInfo[i].cancel();
        } //-----
        byte selCount = 0; // Επιβεβαίωση επιλογής κουτιού -----
        for (int i = 0; i < 5; i++)
            if (btnBox[i].isSelected()) {
                selCount++;
                if (boxNo != i+1) btnBox[i].cancel();
            } if (selCount == 0) {
                boxNo = 0;
                status = NONE; }
        //-----
        if (btnSelect.isPressed(scaledX, scaledY)) { //continue with selected box
            if (box[boxNo-1].dikaioma) { epilogiKategorias();

```

```

        break; }
        dinetaiEnigma((byte)0);
        if (box[boxNo-1].periexomeno == Box.Bonus.SFIGGES)
            dinetaiEnigma((byte)0);
    } else btnSelect.cancel();
}
return true;
}

//-----
// renderBox
//-----
private void renderBox(Grafika grafika) {
    for (UIButton btn : btnBox) btn.render(grafika); }

//-----
// renderPrompt
//-----
private void renderPrompt(Grafika grafika) {
    grafika.setColor(Color.rgb(102, 40, 0));
    grafika.setTextSize(12);
    grafika.drawString("ΕΠΕΛΕΞΕ ΚΟΥΤΙ"
        , (Platform.GAME_WIDTH - grafika.getStringWidth("ΕΠΕΛΕΞΕ ΚΟΥΤΙ")) / 2
        , titleTop);
}

//-----
// renderInfo
//-----
private void renderInfo(Grafika grafika) {
    btnSelect.render(grafika);
    grafika.setColor(Color.rgb(102, 40, 0));
    grafika.setTextSize(12);
    grafika.drawString("ΠΕΡΙΕΧΟΜΕΝΟ " + boxNo + "ου ΚΟΥΤΙΟΥ", titleLeft, titleTop);
    grafika.setTextSize(10);
    grafika.drawString(box[boxNo - 1].getTitleInfo(), 10
        , titleTop + grafika.getStringHeight(box[boxNo-1].getTitleInfo()) + 10);
    txtBoxInfo[boxNo-1].render(grafika);
    grafika.setColor(Color.argb(255, 255, 255, 255));
}

//-----
// renderBoxContent
//-----
private void renderBoxContent(Grafika grafika) {
    Bitmap bmp;
    int left, top;

    switch (box[boxNo-1].periexomeno) {
    case SFIGGA:
    case SFIGGA9: bmp = Assets.sphinx[0];
        break;
    case SFIGGES: bmp = Assets.sphinx[0];
        left = btnBox[boxNo-1].getX()
            + Assets.boxOpen.getWidth()/2 - 2*bmp.getWidth()/3;
        top = btnBox[boxNo-1].getY() - 3*bmp.getHeight()/4;
        grafika.drawImage(bmp, left, top);
        left += bmp.getWidth()/3;
        grafika.drawImage(bmp, left, top);
        return;
    case ZARI: bmp = Assets.diceIsometric;
        break;
    case TIPOTA:
    default: return;
    }
    left = btnBox[boxNo-1].getX()+(Assets.boxOpen.getWidth()-bmp.getWidth())/2;
    top = btnBox[boxNo-1].getY() - 3*bmp.getHeight()/4;

    grafika.drawImage(bmp, left, top);
}

```

```

//-----
// renderCategories
//-----
private void renderCategories(Grafika grafika) {
    String str = "ΕΠΕΛΕΞΕ ΚΑΤΗΓΟΡΙΑ";
    grafika.setFont(Assets.comicsans, 12);
    int x = (Platform.GAME_WIDTH - grafika.getStringWidth(str)) / 2;
    int y = Platform.GAME_HEIGHT / 3;
    grafika.setColor(Color.rgb(102, 40, 0)); //BROWN
    grafika.drawString(str, x, y);

    for (UIButton btnCategory : txtBoxInfo) btnCategory.render(grafika);
}

//-----
// renderZariResult
//-----
private void renderZariBonus(Grafika grafika) {
    String strZariTitle, strZariBonus;
    int arithmos = zari.getArithmos();

    switch (arithmos) {
    case 1:
    case 2: strZariTitle = "ΔΥΣΤΥΧΩΣ...";
            strZariBonus = "Έχασες 1 ζωή!";
            break;
    case 3:
    case 4: strZariTitle = "ΔΕΝ ΠΕΙΡΑΖΕΙ";
            strZariBonus = "Δεν κέρδισες ούτε έχασες τίποτα.";
            break;
    case 5: strZariTitle = "ΜΠΡΑΒΟ!";
            strZariBonus = "Κέρδισες 1 ζωή!";
            break;
    case 6:
    default: strZariTitle = "ΣΥΓΧΑΡΗΤΗΡΙΑ!!";
            strZariBonus = "Κέρδισες 2 ζωές!!!";
    }
    grafika.setColor(Color.rgb(102, 40, 0));

    grafika.setFontDefault(14); //προβολή χαρακτηρισμού αποτελέσματος
    titleLeft = (Platform.GAME_WIDTH-grafika.getStringWidth(strZariTitle)) / 2;
    titleTop = Platform.GAME_HEIGHT / 3;
    grafika.drawString(strZariTitle, titleLeft, titleTop);

    grafika.setTextSize(20); //προβολή αριθμού αποτελέσματος ζαριού
    int left = (Platform.GAME_WIDTH-grafika.getStringWidth("6")) / 2;
    int top = Platform.GAME_HEIGHT / 2;
    grafika.drawString(String.valueOf(arithmos), left, top);

    grafika.setTextSize(14); //προβολή περιγραφής αποτελέσματος
    left = (Platform.GAME_WIDTH-grafika.getStringWidth(strZariBonus)) / 2;
    top = 2 * Platform.GAME_HEIGHT / 3;
    grafika.drawString(strZariBonus, left, top);

    grafika.setTextSize(11);
    grafika.drawString("Ακουμπήστε την οθόνη για συνέχεια στο λαβύρινθο."
        , 10, Platform.GAME_HEIGHT - 10);
}

//-----
// dinetaiEnigma
//-----
private void dinetaiEnigma(byte category) {
    status = END;
    if (category == 0) category = (byte)(new Random().nextInt(5));

    switch (box[boxNo-1].periexomeno) {
    case SFIGGES: t = 2;
    case SFIGGA:

```

```

        sfigga = new Sfigga((byte)4, category, (byte)0);
        break;
    case SFIGGA9: sfigga = new Sfigga((byte)9, category, (byte)0);
        break;
    case ZARI:
        zari = new Zari(this);
        setCurrentState(zari);
    default: return;
    }
    sfigga.kaneiErotisi(this, player);
}

//-----
// epilogiKategorias
//-----
private void epilogiKategorias() {
    int left = 4;
    int top = Platform.GAME_HEIGHT/2;
    int columnWidth = (Platform.GAME_WIDTH-8) / 5;
    int width = columnWidth - 3;

    for (int i = 0; i < 5; i++, left+=columnWidth) {
        txtBoxInfo[i]=new UIButton(left, top, width, Basi.getCategories(i), 10
            , Color.rgb(200, 200, 175), Color.rgb(102, 40, 0), Color.BLACK);
    }
    status = CATEGORY;
}

//-----
// kataxorisiBonus
//-----
private void kataxorisiBonus() {
    switch (box[boxNo-1].periexomeno) {
    case TIPOTA:
        setCurrentState(new Labyrinth(player, player.icon));
        break;
    case ZARI:
        zariBonus();
        status = ZARI;
        break;
    default:
        int result = sfigga.getApotelesma();
        if (t == 0) {
            if (result > 0) {
                Score.athroizetaiProspath();
                if (box[boxNo - 1].periexomeno == Box.Bonus.SFIGGA9)
                    Score.athroizetaiProspath();
            } else Score.xaneiProspath();
            setCurrentState(new Labyrinth(player, player.icon));
        } else {
            if (result > 0) Score.athroizetaiProspath();
            dinetaiEnigma((byte) 0);
            t = 0;
        }
    }
}

private void zariBonus() {
    int arithmos = zari.getArithmos();

    switch (arithmos) {
    case 1:
    case 2: Score.xaneiProspath();
        break;
    case 6: Score.athroizetaiProspath();
    case 5: Score.athroizetaiProspath();
        break;
    default:
    }
}

```

```
}  
}
```

Γ.32 Αρχείο Zari.java

```
/**  
// Java Class Name : Zari  
**/  
package grpekast.sch.users.state;  
  
import android.graphics.Color;  
import android.view.MotionEvent;  
  
import java.util.Random;  
  
import grpekast.sch.users.labyrinthos.Assets;  
import grpekast.sch.users.labyrinthos.Platform;  
import grpekast.sch.users.utility.Grafika;  
  
public class Zari extends State {  
    private final int ZARI_LENGTH;  
    private final float ACCEL = -128f;  
    private float Ux = 1, Uy = 1; // Velocity at X, Y  
    private float X, Y, oldX, oldY;  
    private float S = 0;  
    private int arithmos = 0;  
    private Random rand = new Random();  
    private boolean stopped = false, rolling = false, clearMessage = false;  
    private State sourceState;  
  
    //-----  
    // Constructor  
    //-----  
    public Zari(State state) {  
        sourceState = state;  
        ZARI_LENGTH = Assets.dice[0].getWidth();  
        X = (Platform.GAME_WIDTH - ZARI_LENGTH) / 2;  
        Y = (Platform.GAME_HEIGHT - ZARI_LENGTH) / 2;  
    }  
  
    //-----  
    // riximo  
    //-----  
    public void riximo(float ax, float ay) {  
        Ux = Math.abs(ax) * 50 * Math.signum(Ux);  
        Uy = Math.abs(ay) * 50 * Math.signum(Uy);  
        if (!rolling) {  
            Assets.playSound(Assets.diceID);  
            rolling = true; clearMessage = true;  
            resumeThread();  
        }  
    }  
  
    //-----  
    // setBackground  
    @Override //-----  
    public void setBackground(Grafika grafika) {  
        grafika.clear(Color.BLACK);  
        grafika.setColor(Color.WHITE);  
        grafika.drawString("Κούνησε τη συσκευή για να ρίξεις το Ζάρι.", 2, 16);  
        grafika.drawImage(Assets.dice[arithmos], (int) X, (int) Y  
            , ZARI_LENGTH, ZARI_LENGTH);  
        grafika.setColor(Color.BLACK);  
    }  
}
```

```

//-----
// update
@Override //-----
public void update(float dt) {
    if (rolling) {
        if (Math.abs(Ux) < 5 && Math.abs(Uy) < 5) {
            S = 0;
            rolling = false;
            stopped = true;
            pauseThread();
        }

        double U = Math.sqrt(Ux*Ux + Uy*Uy); //Συνιστάμενη ταχύτητα
        S += (U * dt + ACCEL*dt*dt/2);
        if (S > 16) { // Νέος αριθμός ζαριού
            arithmos = rand.nextInt(6);
            S = 0;
        }

        if (X < 0) Ux = Math.abs(Ux);
        else if (X > Platform.GAME_WIDTH-ZARI_LENGTH) Ux = -Math.abs(Ux);
        else if (Y < 0) Uy = Math.abs(Uy);
        else if (Y > Platform.GAME_HEIGHT-ZARI_LENGTH) Uy = -Math.abs(Uy);

        oldX = X;
        oldY = Y;
        X += Ux*dt + ACCEL*dt*dt/2 * Math.signum(Ux);
        Y += Uy*dt + ACCEL*dt*dt/2 * Math.signum(Uy);

        Ux += ACCEL*dt * Math.signum(Ux);
        Uy += ACCEL*dt * Math.signum(Uy);
    }
}

//-----
// render
@Override //-----
public void render(Grafika grafika) {
    if (clearMessage) {
        grafika.fillRect(0, 0, Platform.GAME_WIDTH, 20);
        clearMessage = false;
    }
    if (rolling) {
        grafika.fillRect((int) oldX, (int) oldY, ZARI_LENGTH, ZARI_LENGTH);
        grafika.drawImage(Assets.dice[arithmos], (int) X, (int) Y
            , ZARI_LENGTH, ZARI_LENGTH);
    }
}

//-----
// onTouch
@Override //-----
public boolean onTouch(MotionEvent myEvent, int scaledX, int scaledY) {
    if (rolling) return false;

    switch (myEvent.getAction()) {
        case MotionEvent.ACTION_DOWN:
            return true;
        case MotionEvent.ACTION_UP:
            if (stopped) setCurrentState(sourceState);
    }
    return false;
}

//-----
// onSensor
@Override //-----
public void onSensor(float ax, float ay) {
    if (stopped) return;

    if (!rolling) {

```

```

        if (Math.abs(ax) > 12 || Math.abs(ay) > 12) riximo(ay, ax); }
    else {
        if (ax > 10 || ax < -10 || ay > 10 || ay < -10) riximo(ay, ax); }
    }

    //-----
    // getArithmos
    //-----
    public int getArithmos() { return arithmos + 1; }
}

```

Γ.33 Αρχείο GameOver.java

```

//*****
// Java Class Name : GameOver
//*****
package grpekast.sch.users.state;

import android.graphics.Color;
import android.graphics.Typeface;
import android.view.MotionEvent;

import grpekast.sch.users.labyrinthos.Assets;
import grpekast.sch.users.labyrinthos.Platform;
import grpekast.sch.users.model.Score;
import grpekast.sch.users.utility.Grafika;

public class GameOver extends State {
    private int playerImageIndex;
    private String playerScore;
    private boolean playerIsAlive;
    private final String[] strRank = new String[] { "ΤΕΛΟΣ ΠΑΙΧΝΙΑΙΟΥ",
                                                    "ΣΥΓΧΑΡΗΤΗΡΙΑ ΒΡΗΚΕΣ ΤΗΝ ΕΞΟΔΟ!!" };

    //-----
    // Constructor
    //-----
    public GameOver(int pIndex) {
        playerImageIndex = pIndex;
        int prospatheia = Score.getProspatheia();
        playerIsAlive = (prospatheia >= 0);

        if (playerIsAlive) {
            mIntroFile = "ppanther_cut3.mp4";
            Score.katagrafetai(5 * prospatheia);
        }
        playerScore = Score.anagrafetai();
    }

    //-----
    // setBackground
    @Override //-----
    public void setBackground(Grafika grafika) {
        grafika.reset();
        render(grafika);
        if (playerIsAlive) Assets.playSound(Assets.labyrinthosID);
        else Assets.playSound(Assets.gameOverID);
    }

    //-----
    // update
    @Override //-----
    public void update(float delta) { }

    //-----
    // render
    @Override //-----

```



```

public void render(Grafika grafika) {
    grafika.setColor(Color.rgb(255, 145, 0));
    grafika.fillRect(0, 0, Platform.GAME_WIDTH, Platform.GAME_HEIGHT);
    grafika.setColor(Color.DKGRAY);
    grafika.setFont(Typeface.DEFAULT_BOLD, 18);

    if (playerIsAlive)
        renderRank(grafika);
    else
        renderGameOver(grafika);
}

//-----
// onTouch
@Override //-----
public boolean onTouch(MotionEvent e, int scaledX, int scaledY) {
    if (e.getAction() == MotionEvent.ACTION_UP) {
        if (playerIsAlive) playerIsAlive = false;
        else {
            Platform myActivity = (Platform)Platform.sGame.getContext();
            myActivity.onBackPressed();
        }
    }
    return true;
}

private void renderRank(Grafika grafika) {
    String str = strRank[1];
    int x = (Platform.GAME_WIDTH - grafika.getStringWidth(str)) / 2;
    int y = Platform.GAME_HEIGHT / 4;
    grafika.drawString(str, x, y);

    int width = Platform.GAME_WIDTH / 8;
    x = (Platform.GAME_WIDTH - width) / 2;
    y = (Platform.GAME_HEIGHT - width) / 2;
    grafika.drawImage(Assets.beastie[playerImageIndex], x, y, width, width);

    byte prospatheiaCount = Score.getProspatheia();
    width = Assets.heart.getWidth();
    x = (Platform.GAME_WIDTH
        -prospatheiaCount*width-grafika.getStringWidth(" x 5")) / 2;
    y = 2 * Platform.GAME_HEIGHT / 3;
    for (byte i = 0; i < prospatheiaCount; i++, x+=width)
        grafika.drawImage(Assets.heart, x, y);
    y += grafika.getStringHeight("5");
    grafika.drawString(" x 5", x, y);

    str = (5*prospatheiaCount) + " Επιπλέον πόντοι";
    x = (Platform.GAME_WIDTH - grafika.getStringWidth(str)) / 2;
    y += grafika.getStringHeight("5") + 5;
    grafika.drawString(str, x, y);

    grafika.setTextSize(11);
    grafika.drawString("Ακουμήστε την οθόνη.", 10, Platform.GAME_HEIGHT - 10);
}

private void renderGameOver(Grafika grafika) {
    String str = strRank[0];
    int x = (Platform.GAME_WIDTH - grafika.getStringWidth(str)) / 2;
    int y = (Platform.GAME_HEIGHT - 12 - (grafika.getStringHeight(str)
        + grafika.getStringHeight(playerScore))) / 2;
    grafika.drawString(str, x, y);

    str = "ΕΚΟΡ " + playerScore;
    x = (Platform.GAME_WIDTH - grafika.getStringWidth(str)) / 2;
    y += grafika.getStringHeight(str) + 6;
    grafika.drawString(str, x, y);
}
}

```

Βιβλιογραφία

- [01] «Android, the world's most popular mobile platform». Android Developers, Accessed 29 November 2014. <http://developer.android.com/about/index.html>.
- [02] Marc Prensky. «Digital game-based learning». New York: McGraw Hill, pp. 106-107, 2001.
- [03] Statista. «Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 4th quarter 2013». The Statistics Portal, Statista.com, 2014. <http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.
- [04] «Games for Local Wi-fi(LAN) or Bluetooth multiplayer». xda-developers, Accessed 29 November 2014. <http://forum.xda-developers.com>.
- [05] ELECTRONIC ARTS. «SCRABBLE». Google Play, Last modified 27 November 2014. <https://play.google.com/store/apps/details?id=com.ea.game.scrabblemattel> by.
- [06] Zynga. «New Words With Friends». Google Play, Last modified 28 November 2014. <https://play.google.com/store/apps/details?id=com.zynga.wwf2.free>.
- [07] Angry Mob Games. «Guerrilla Bob». Google Play, Last modified 13 October 2013. <https://play.google.com/store/apps/details?id=com.angrymobgames.guerrillabob>.
- [08] educ8s.com. «Γίνε πλούσιος». Google Play, Last modified 5 November 2014. <https://play.google.com/store/apps/details?id=com.ekatommyriouxos>.
- [09] cool cherry trees. «4 Player Reactor (Multiplayer)». Google Play, Last modified 27 June 2014. <https://play.google.com/store/apps/details?id=coolcherrytrees.games.reactor4>.
- [10] Optimus Prime Apps. «Ποιος το ξέρει;». Google Play, Last modified 18 August 2014. <https://play.google.com/store/apps/details?id=com.mrpdev.apps.games.mmo&hl=el>.

- [11] mobirix. «Ο βασιλιάς του λαβυρίνθου». Google Play, Last modified 17 November 2014. <https://play.google.com/store/apps/details?id=com.mobirix.playmaze>.
- [12] mobirix. «Ο βασιλιάς tic-tac-toe». Google Play, Last modified 25 November 2014. https://play.google.com/store/apps/details?id=com.mobirix.ttt_gg.
- [13] Quizify. «IT Logic Quiz». Google Play, Last modified 20 October 2014. <https://play.google.com/store/apps/details?id=com.itlogical.unique.quiz.app.game.puzzle>.
- [14] M.D. Roblyer. «Εκπαιδευτική Τεχνολογία και Διδασκαλία». Εκδόσεις Έλλην (ΙΩΝ), 2008.
- [15] Diana Laurillard. «Rethinking University Teaching: A framework for the effective use of educational technology». London: Routledge, 1993.
- [16] Diana Laurillard. «Rethinking University Teaching. A conversational framework for the effective use of learning technologies». London: Routledge, 2002.
- [17] Gilbert Lester, Gale Veronica. «Principles of E-learning Systems Engineering». Chandos Series for Information Professionals, pp 1-187, 2005.
- [18] Android Developers. «Android APIs - Package Index». Android Developers, Last accessed 5 September 2015. <https://developer.android.com/reference/packages.html>.
- [19] Oracle and/or its affiliates. «Java SE Downloads». Oracle Technology Network, Last accessed 5 September 2015. <http://www.oracle.com/technetwork/java/javase/downloads/>.
- [20] Modelio.org community and others. «Modelio - the open source modeling environment». Modelio Open Source Community, Last accessed 5 September 2015. <https://www.modelio.org/>.
- [21] Google. «Android Studio - The official Android IDE». Android Developers, Last accessed 5 September 2015. <https://developer.android.com/sdk/index.html>.
- [22] Oracle Corporation. «VirtualBox». VirtualBox.org, Last accessed 5 September 2015. <https://www.virtualbox.org/>.

- [23] dotPDN LLC and Rick Brewster. «Paint.NET - Free Software for Digital Photo Editing». paint.net, Last accessed 5 September 2015. <http://www.getpaint.net/index.html>.
- [24] ButtonBass. «ButtonBass Player Piano». ButtonBass.com, Last accessed 5 September 2015. <http://www.buttonbass.com/PlayerPiano.html>.
- [25] Audacity Team. «Audacity - Free Audio Editor and Recorder». Members of the Audacity development team, Last accessed 5 September 2015. <http://audacityteam.org/>.
- [26] DB Browser for SQLite. «DB Browser for SQLite». sqlitebrowser.org, Last accessed 5 September 2015. <http://sqlitebrowser.org/>.
- [27] Dropbox. «Dropbox». dropbox.com, Last accessed 5 September 2015. <https://www.dropbox.com/>.
- [28] Abbott, J. Russell. «Program design by informal English descriptions». Communications of the ACM Volume 26 Issue 11, pp. 882-894, 1983.
- [29] Dr. Jürgen Pitschke. «Textual Analysis – A Working Technique for Visual Modeling». BCS - Dr. Jürgen Pitschke, Version 2.0, 2012.
- [30] James S. Cho. «The Beginner's Guide to Android Game Development». Glasnevin Publishing, 2014.
- [31] Bill Phillips, Brian Hardy. «Android Programming: The Big Nerd Ranch Guide». Pearson Technology Group, 2013.
- [32] The Stack Overflow community. «StackOverflow». Stack Exchange network of Q&A sites. <http://stackoverflow.com/>.