Τ. Ε. Ι. ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΛΑΤΦΟΡΜΑΣ ΜΑΘΗΜΑΤΙΚΩΝ ΔΗΜΟΤΙΚΟΥ ΣΧΟΛΕΙΟΥ

Δεσύλλας Δημήτριος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων
ΠΕΤΡΟΣ ΛΑΜΨΑΣ
ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Λαμία 2013

ΕΥΧΑΡΙΣΤΙΕΣ

Η ολοκλήρωση αυτής της πτυχιακής υλοποιήθηκε με την υποστήριξη ενός αριθμού ανθρώπων στους οποίους θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου.

Πρώτα από όλους θα ήθελα να ευχαριστήσω την Τετάρτη τάξη του 2ου Δημοτικού Σχολείου Ατσιποπούλιου στην Κρήτη, που οι μαθητές δοκίμασαν την έκδοση 1.6.1 του guma.

Ακόμη ξεχωριστές ευχαριστίες στον δάσκαλό του Χρυσόστομο Καραβούνη που έλαβε την προτοβουλία όσο να το δοκιμάσει τόσο στο 2ου Δημοτικού Σχολείου Ατσιποπούλιου που δίδασκε αυτήν την περίοδο όσο και στο σχολείο που διδάσκει αυτήν την στιγμή. Καθώς και για την προώθησή του project στους συναδέλφους του.

Το blog του osarena (www.osarena.net) που έχει κάνει ένα άρθρο (http://osarena.net/logismiko/applications/guma-efarmogi-arithmitikis-gia-mathites-tou-dimotikou.html) για την έκδοση 1.6.1. Καθώς και την συγγραφέα του άρθρου Έλενα.

Τον συμφοιτητή μου Δημήτριο Διαμαντή για κάποιες ιδέες που μου έδωσε όσο και στην παρούσα φάση του project όσο και τις ιδέες του για μελλοντικές τροποποιήσεις και προσθήκες.

Τέλος τον κύριο Σίμο Ξενιτέλη, μέλος της Ελληνικής κοινότητας Ubuntu-gr, που μου έκανε το αρχικό makefile της εφαρμογής πριν ακόμη αναπτυχθεί μέσω αυτής της πτυχιακής.

Δεσύλλας Δημήτριος,
Σεπτέμβριος 2013

ΠΕΡΙΛΗΨΗ

Το guma είναι μια μικρή εκπαιδευτική εφαρμογή μαθηματικών για μαθητές δημοτικού. Η εφαρμογή είναι ελευθέρου λογισμικού υπό την άδεια GNU GPL v3. Η φιλοσοφία πίσω από αυτό είναι ότι ο μαθητής – χρήστης δίνει το αποτέλεσμα μιας αριθμητικής πράξης και σε περίπτωση λάθους δείχνει βήμα βήμα πως γίνεται η πράξη αυτή. Είναι desktop application και έχει αναπτυχθεί σε java.

Στην πτυχιακή θα καλύψουμε τα εξής θέματα:

- Γενικά για τη εφαρμογή (η φιλοσοφία της εφαρμογής και ο λογικός Διαχωρισμός).
- Το κυρίως πρόγραμμα. (φιλοσοφία, ροή εκτέλεσης)
- Ο προσομοιωτής. (φιλοσοφία, ροή εκτέλεσης)
- Δυνατότητα να ανοίγει παιχνίδι από το διαδίκτυο. (φιλοσοφία, ροή εκτέλεσης)
- Τα πακέτα και τις κλάσεις του Guma.
- Μελλοντικός σκοπός του project.

Ο κώδικας της εφαρμογής είναι διαθέσιμος μέσω github (https://github.com/pc-magas/guma) ενώ ο κώδικας της τελευταίας σταθερής έκδοσης όπως και το μεταγλωττισμένο πρόγραμμα στις τελευταίες εκδόσεις είναι στο sourceforge (https://sourceforge.net/projects/guma-efs).

ΠΕΡΙΕΧΟΜΕΝΑ

1 ο ΚΕΦΑΛΑΙΟ

ΓΕΝΙΚΑ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ

1 Εισαγωγή:

Όπως ανέφερα στην περίληψη το Guma είναι μια εκπαιδευτική πλατφόρμα ελεύθερου λογισμικού (Άδεια GNU GPL v3). Το όνομά του προέρχεται από το συνδυασμό των λέξεων Guess και Math.

Σε γενικές γραμμές αυτό που κάνει είναι να εμφανίζει κάποιες βασικές αριθμητικές πράξεις και ο μαθητής να καλείτε να μαντέψει το αποτέλεσμα των πράξεων αυτών και σε περίπτωση λάθους του δείχνει βήμα - βήμα πως εκτελείτε η πράξη αυτή.

2 Γλώσσα προγραμματισμού - τεχνολογίες

Η γλώσσα η οποία είναι γραμμένη είναι η Java. Και την επέλεξα τους εξής λόγους λόγους:

- Είναι platform indepedent δηλαδή εξαρτάτε από μια εικονική μηχανή για να τρέξει και όχι απ' ευθείας στο Λειτουργικό σύστημα.

- Δεν είναι time critical εφαρμογή δηλαδή δεν με ένοιαζε τόσο πολύ η ταχύτητα που θα εκτελούταν.
- Ήταν η μόνη γλώσσα που είχε build in κλάσεις και μεθόδους για γραφικό περιβάλλον. (Java Swing)
- Έχει δυνατότητα να τρέξει και στο web είτε μέσω servlet και JSP τεχνολογιών είτε μέσω Applet (που πλέον δεν είναι τόσο διαδεδομένα).

Όμως η χρήση java έχει ένα μειονέκτημα, είναι σχετικά "βαριά" και πολλά σχολεία ενδεχομένως να μην έχουν αρκετά καλές υποδομές προκειμένου να τρέξει.

3 Χαρακτηριστικά Εφαρμογής:

Η εφαρμογή διαθέτει τα εξής κυρίως χαρακτηριστικά:

1. Δυνατότητα να δημιουργεί τυχαίες πράξεις 2 αριθμών βάση 3 παραμέτρων:
    1. Το Πλήθος των πράξεων
    2. Την μέγιστη τιμή κάθε όρου μιας πράξης
    3. Το είδος της πράξης δηλαδή αν είναι Πολλαπλασιασμός, Διαίρεση, Πρόσθεση ή Αφαίρεση. Ο χρήστης μπορεί να επιλέξει παραπάνω από ένα είδος.
2. Δυνατότητα προσομοιώνει την εκάστοτε πράξη σε περίπτωση λάθους. Δηλαδή να δείχνει αναλυτικά πως γίνεται μια πράξη όπως θα την εκτελούσε ο μαθητής στο χαρτί. Έτσι δίνεται η δυνατότητα να κατανοήσει καλύτερα την εκάστοτε πράξη που έκανε λάθος
3. Δυνατότητα ανοίγματος ενός αποθηκευμένου παιχνιδιού που είναι ανεβασμένο στο web απ' ευθείας από την εφαρμογή. Με απλά λόγια αν ένας χρήστη έχει ανεβάσει σε μια cloud εφαρμογή ή στον ιστότοπό του

ένα αρχείο .guma να μπορεί δίνοντας μόνο το url του αρχείου από το web να μπορεί να το ανοίγει.

4 Λογικός Διαχωρισμός Εφαρμογής για μελέτη:

Για να γίνει ποιο κατανοητή η εφαρμογή την διαχωρίζουμε σε 3 τμήματα:

1. Στην *κυρίως εφαρμογή*: Που είναι το τμήμα που ο χρήστης δημιουργεί ένα νέο παιχνίδι. Και σε αυτό προσπαθεί να βρει στην εκάστοτε πράξη το σωστό αποτέλεσμα.

2. Στον *προσομοιωτή πράξεων*: Που ο χρήστης έχει την δυνατότητα να δει πως εκτελείτε βήμα βήμα η πράξη.

3. Στον *μηχανισμό φόρτωσης παιχνιδιού από το web*: Που ο χρήστης έχει την δυνατότητα να ανοίξει ένα παιχνίδι που είναι αποθηκευμένο στο web (μέσω cloud storage ή σε ένα ιδιόκτητο http server).

Στα επόμενα μέρη αυτής της πτυχιακής θα σας αναλύσω αναλυτικά και με διαγράμματα UML τι κάνει κάθε μέρος και πως λειτουργεί.

5 Επίλογος

Αυτό το κεφάλαιο αποτελεί μαι γενική εισαγωγή, που περιγράφει τι χαρακτηριστικά έχει το guma. Ακόμα περιγράφω σε ποια λογικά μέρη έχω διασπάσει την εφαρμογή μου προκειμένου να την αναλύσω και να την περιγράψω όσο ποιο πλήρες και αναλυτικά μπορώ.

2 ο ΚΕΦΑΛΑΙΟ

Η ΚΥΡΙΩΣ ΕΦΑΡΜΟΓΗ

1 Εισαγωγή

Εδώ θα μελετήσουμε το κυρίως μέρος της εφαρμογής δηλαδή το μέρος που ο χρήστης δημιουργεί ένα νέο παιχνίδι που σε αυτό προσπαθεί να βρει το αποτέλεσμα μιας πράξης, που αποτελεί και τον βασικό κορμό αλλά και την αρχική ιδέα της εφαρμογής.

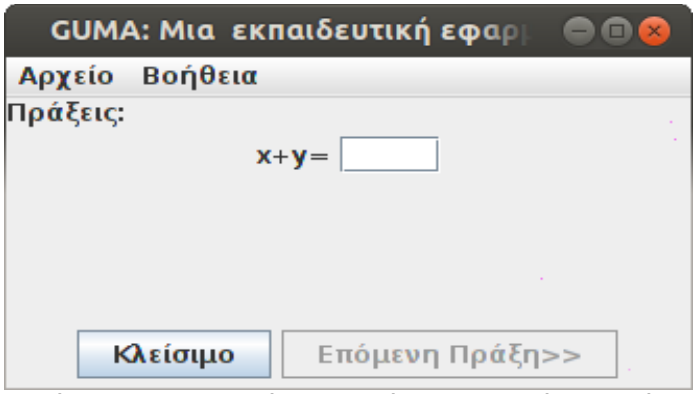
Θα δούμε τον βασικό Αλγόριθμο καθώς και διαγράμματα UML (Σπύρου, 2011-2012) που περιγράφουν την λειτουργία αυτού του μέρους της εφαρμογής.

2 Το Περιβάλλον της κύριας εφαρμογής

Η κυρίως εφαρμογή αποτελείτε κυρίως από 2 βασικά παράθυρα:

1. Το κυρίως παράθυρο που είναι το παράθυρο που ο Χρήστης δίνει το αποτέλεσμα μιας Πράξης.
2. Το παράθυρο νέου παιχνιδιού που ο χρήστης ορίζει παραμέτρους για ένα νέο παιχνίδι.

2.1 Το κυρίως Παράθυρο.

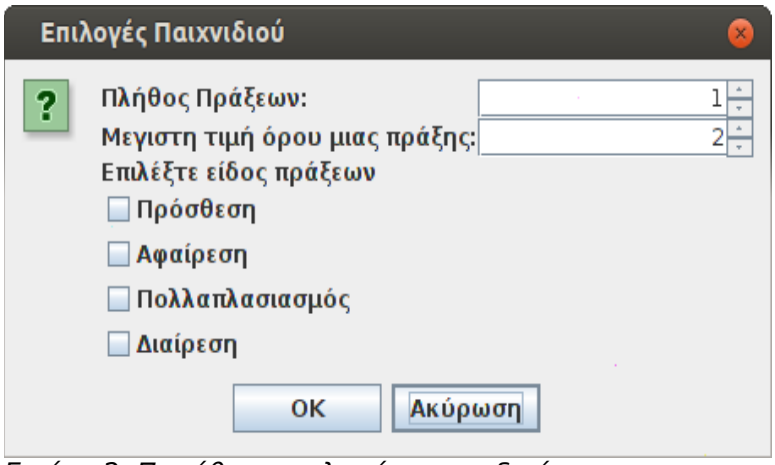


Εικόνα 1: Το κυρίως παράθυρο χωρίς να έχει ξεκινήσει νέο παιχνίδι

Το κυρίως παράθυρο (Εικόνα 2) περιέχει έχει 2 μενού:

1. Αρχείο που περιέχει λειτουργίες όπως:
    1. Νέο Παιχνίδι: Άνοιγμα παραθύρου που δημιουργεί ένα νέο παιχνίδι.
    2. Αποθήκευση, Αποθήκευση ως: Αποθηκεύει την πρόοδο σε ένα αρχείο xml.
    3. Άνοιγμα: Ανοίγει την αποθηκευμένη πρόοδο.
    4. Άνοιγμα από το διαδίκτυο: Σας δίνει την δυνατότητα να ανοίγεται ένα παιχνίδι που είναι αποθηκευμένο στο διαδίκτυο. Έτσι μπορείτε να έχετε ένα αρχείο παιχνιδιού αποθηκευμένο σε ένα cloud storage όπως το UbuntuOne (http:\\one.ubuntu.com) ή το dropbox (http:\\www.dropbox.com) και δίνοντας την διεύθυνση που σας δίνει εφόσον κάνατε το αρχείο δημόσιο να μπορείτε να το ανοίγεται μέσω του GUMA.
2. Και το μενού Βοήθεια που έχει μόνο την Επιλογή: *Σχετικά με το Guma* που εμφανίζει ένα παράθυρο μηνύματος με πληροφορίες σχετικά με την εφαρμογή, την άδεια χρήσης, και τον δημιουργό της.

2.2 Το Παράθυρο επιλογής παιχνιδιού



Εικόνα 2: Παράθυρο επιλογών παιχνιδιού

Το παράθυρο επιλογών παιχνιδιού έχει 3 ρυθμίσεις:

1. Πλήθος Πράξεων:

   Όπως λέει και το όνομά του ο χρήστης επιλέγει σε πόσες πράξεις επιθυμεί να μαντέψει το αποτέλεσμα.

2. Μέγιστη τιμή όρου μιας πράξης:

   Μια πράξη έχει την μορφή *<όρος1> <τελεστής> <όρος2>* με αυτήν την επιλογή ορίζουμε πόσο μεγάλος θα είναι ο *<όρος 1>* και ο *<όρος 2>*.

3. Επιλέξτε το είδος των πράξεων:

   Με αυτήν την επιλογή ο Χρήστης ορίζει τι είδους πράξεων θα εμφανιστούν.

2.3 Άλλα παράθυρα.

Πέρα από το Κυρίως Παράθυρο και το Παράθυρο νέου παιχνιδιού αποτελείτε και από άλλα παράθυρα όπως μηνυμάτων, σφαλμάτων αλλά και παράθυρα ανοίγματος

αρχείων. Αλλά και εισαγωγής επιπλέον αποτελεσμάτων όταν χρειάζεται πχ. Στην διαίρεσή όταν είναι ατελής.

3 Η αρχιτεκτονική του συστήματος.

Η αρχιτεκτονική της κυρίως εφαρμογής είναι Εμπνευσμένη από το Μοντέλο MVC (βλ. Παράρτημα1). Καθώς και το μοντέλο αντικειμενοστραφούς αποδόμησης (Sommerville, 2011, 310).

Θα δούμε παρακάτω ποια είναι η γενική φιλοσοφία λειτουργίας, με απλά λόγια τον βασικό αλγόριθμό που ακολουθά το πρόγραμμα. Ακόμη θα σας δείξουμε πως αλληλεπιδρούν οι Κλάσεις μεταξύ τους μέσω διαγραμμάτων ροής UML. Σε αυτά θα σας δείξουμε για 2 σενάρια - σημεία:

1. Κατά την αρχικοποίηση ενός παιχνιδιού
2. Κατά την κυρίως ροή – διάρκεια ενός παιχνιδιού.

3.1 Κυρίως Αλγόριθμος:

Ο γενικός αλγόριθμός αποτελείτε από τα εξής βήματα:

1. Αρχικοποίηση:
    1. Δημιούργησε μια τυχαία πράξη που οι τελεστές (αριθμοί που συμμετέχουν στην πράξη) να είναι μικρότεροι ή ίσοι από αυτόν που έχει επιλέξει ο χρήστης, και το είδος των πράξεων είναι στο σύνολο που έχει επιλέξει ο χρήστης.
    2. Υπολόγισε το αποτέλεσμα τις πράξης.
    3. Πρόσθεσε το σε μια ArrayList.

4. Επανέλαβε τα παραπάνω βήματα μέχρις ότου φτάσει στο πλήθος πράξεων που έχει επιλέξει ο χρήστης.

2. Για κάθε πράξη:

    1. Ο χρήστης δίνει αποτέλεσμα. Σε περίπτωση Διαίρεσης αν το υπόλοιπο είναι μεγαλύτερο του 0 ο Χρήστης δίνει και το υπόλοιπο.

    2. Αν το αποτέλεσμα ή το υπόλοιπο (όταν χρειάζεται) είναι λάθος:

        1. Βγάλε Μήνυμα λάθους

        2. Μειώνεται ο αριθμός των προσπαθειών.

        3. Ο χρήστης ξαναδίνει το αποτέλεσμα ή το υπόλοιπο (όταν χρειάζεται)

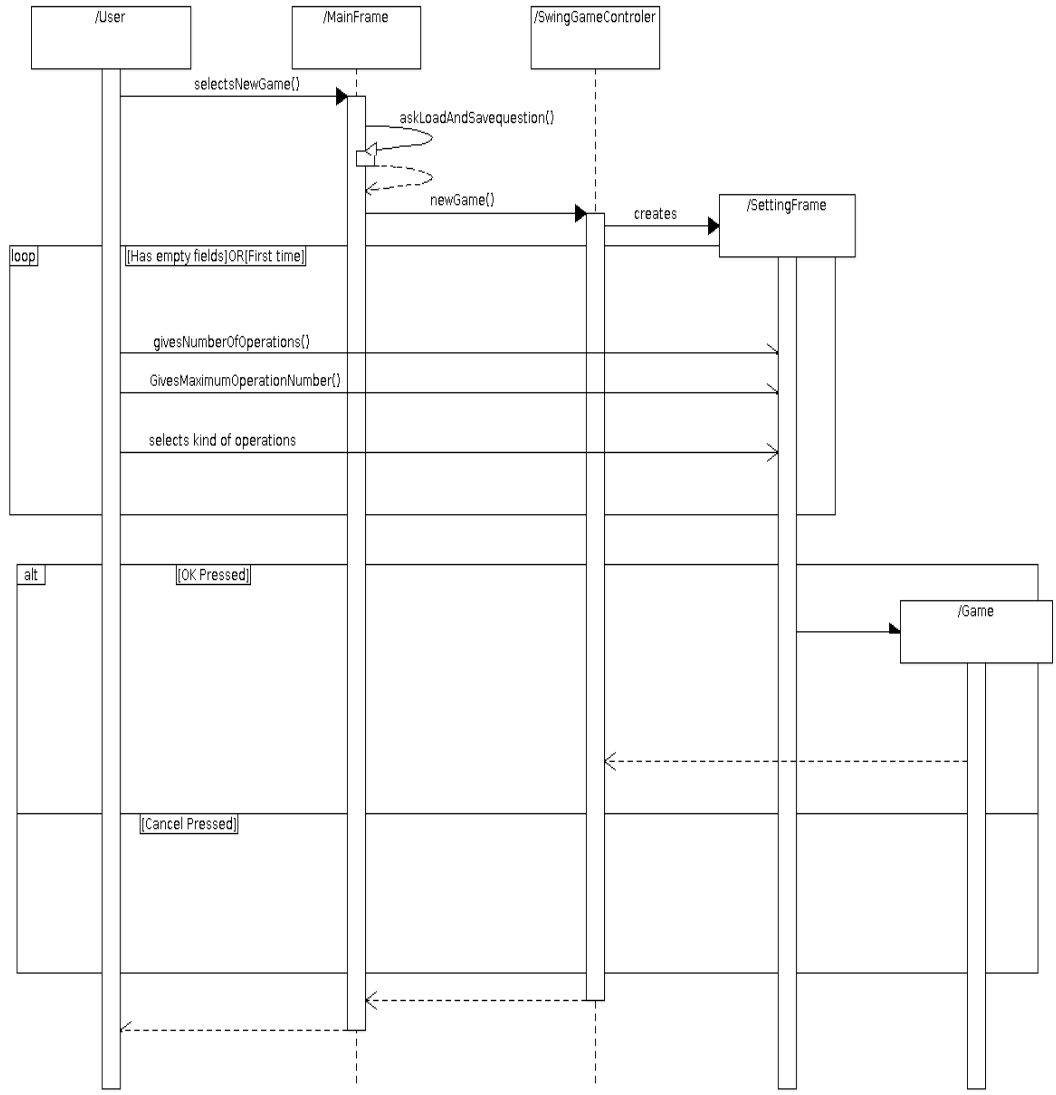
    3. Αν ο Χρήστης δεν εξάντλησε τις προσπάθειές του:

        - Πρόσθεσε τις εναπομείναντες προσπάθειες στο σκορ.

        - Ειδάλλως βγάλε μήνυμα επιλογής για προσομοίωση της πράξης ή μετάβαση στην επόμενη πράξη.

    4. Αν ο Χρήστης έχει άλλες πράξεις να λύσει: Μετάβαση στην επόμενη Πράξη. Ειδάλλως εμφάνισε το σκορ του χρήστη.

3.2 Αλληλεπίδραση Κλάσεων στην Αρχικοποίηση ενός παιχνιδιού.

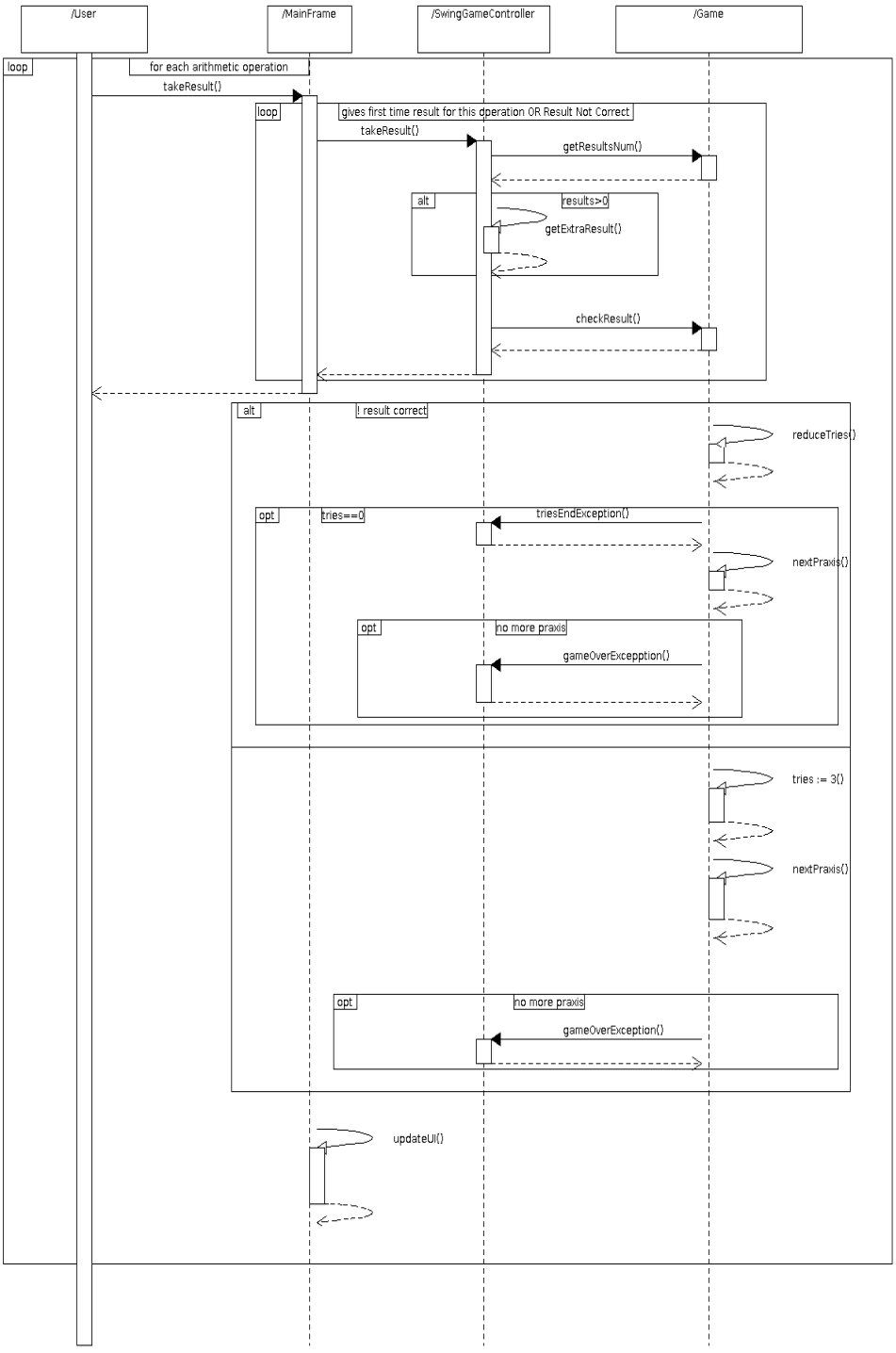


Εικόνα 3: Διάγραμμα ροής κατά την δημιουργία ενός νέου παιχνιδιού

Παρατηρήσεις - Επεξηγήσεις:

- Οι ενέργειες givesNumberOfOperation, givesMaximumOperationNumber και selectskindOfOperations. Δεν αντιστοιχούν σε πραγματικές μεθόδους.
- Η Κλάση Game δεν επιστρέφεται στην κλάση MainFrame.

3.3 Αλληλεπίδραση Κλάσεων στην κύρια ροή ενός παιχνιδιού



Εικόνα 4: Αλληλεπιδράσεις κλάσεων κατά την κύρια ροή ενός παιχνιδιού.

Προϋποθέσεις:

- Ο χρήστης έχει ήδη ξεκινήσει ένα παιχνίδι και έχει κάνει ήδη τις απαραίτητες ρυθμίσεις.

Επεξηγήσεις:

1. Η gameOverException() και οι triesEndException() είναι τα exception που πετά όταν τελειώνουν οι πράξεις και οι προσπάθειες αντίστοιχα και όχι κάποια μέθοδος. Με την gameOverException() τελειώνει το loop εκεί.

2. Ομοίως οι reduceTries() και tries=3() είναι ενέργειες και όχι μέθοδοι της κλάσεις Game αντιστοιχούν στις εντολές tries-- (μείωση των προσπαθειών) και tries=3 (ανανέωση των προσπαθειών σε 3) αντίστοιχα.

3. Η updateUI() ανανεώνει την εμφάνιση του γραφικού περιβάλλοντος.

4 Επίλογος

Σας έδειξα το πως αλληλεπιδρούν οι κλάσεις μέσω διαγραμμάτων UML και πως λειτουργεί το κυρίως πρόγραμμα και πως αλληλεπιδρούν οι κλάσεις μεταξύ τους τόσο στο πως αρχικοποιείται ένα παιχνίδι αλλά και στην κυρία ροή του κυρίως προγράμματος. Ακόμη σας έδειξα το περιβάλλον της εφαρμογής και την βασική ιδέα λειτουργίας του κυρίως προγράμματος.

3 ο ΚΕΦΑΛΑΙΟ

Ο ΠΡΟΣΩΜΟΙΩΤΗΣ ΠΡΑΞΕΩΝ

1 Εισαγωγή

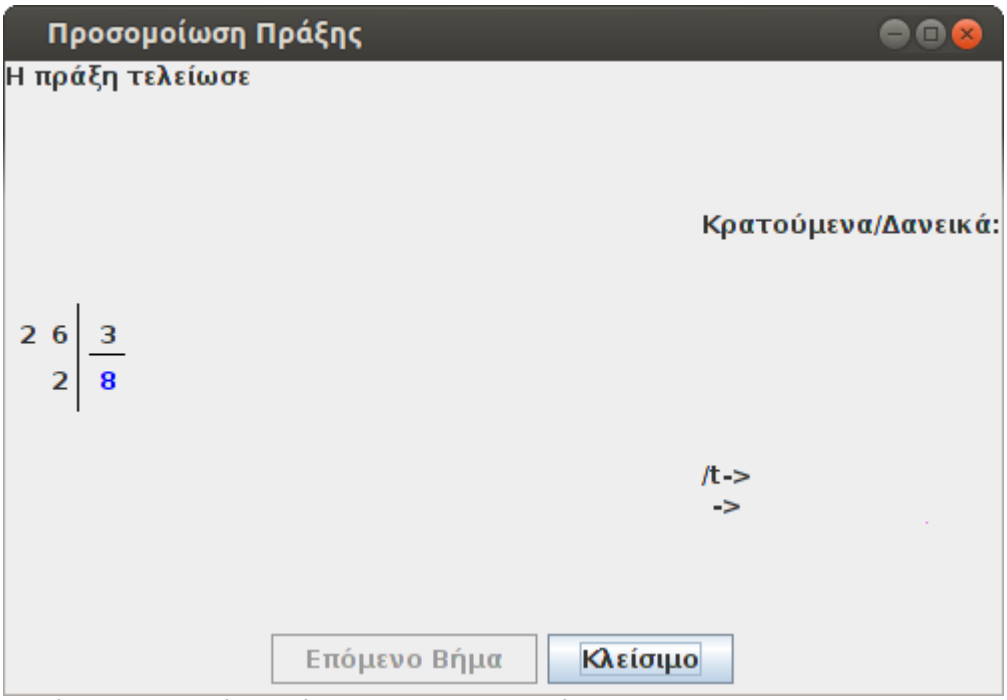
Ένα ιδιαίτερο χαρακτηριστικό είναι ότι εφόσον ο χρήστης εφόσον κάνει λάθος μπορεί το πρόγραμμα να του δείξει βήμα βήμα πως γίνεται η εκάστοτε βασική πράξη που έκανε λάθος. Όπως στο προηγούμενο κεφάλαιο θα σας δείξουμε κάποια βασικά διαγράμματα UML και θα σας εξηγήσουμε την δομή και την λειτουργία του προσομοιωτή.

2 Χαρακτηριστικά

Ο προσομοιωτής ,ανάλογα με την πράξη, εκτελεί τα βήματα που χρειάζεται για να γίνει η πράξη και για κάθε βήμα κρατά και ένα ή και περισσότερα “snapshot” σε μια ArrayList έτσι μετά ο χρήστης το μόνο που έχει να κάνει είναι να βλέπει τα “snapshots”. Τα βήματα της κάθε πράξης είναι τα βήματα που θα εκτελούνταν αν κάναμε την πράξη στο χαρτί.

3 Το περιβάλλον του Προσομοιωτή:

3.1 Βασικό παράθυρο προσομοιωτή:

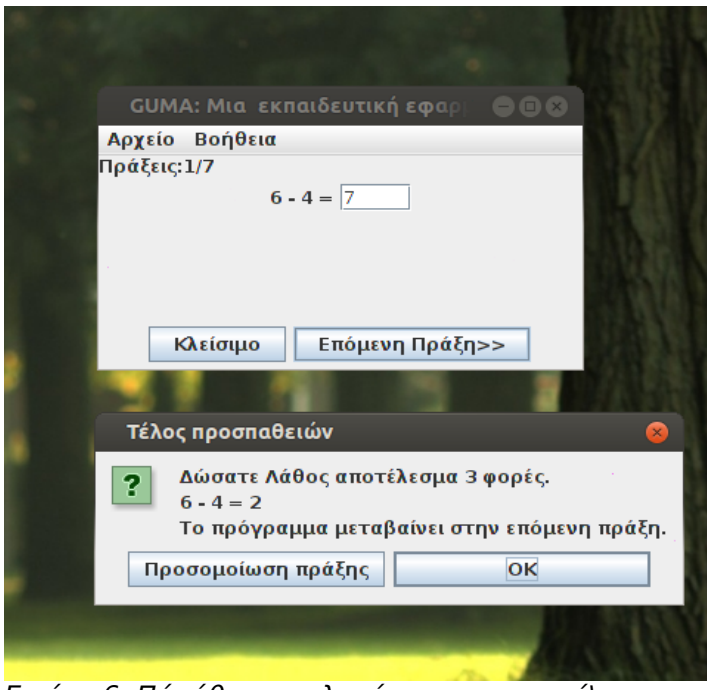


Εικόνα 5: Βασικό Παράθυρο Προσομοιωτή

Το βασικό παράθυρο αποτελείτε από 4 στοιχεία:

1. Στο πάνω μέρος περιέχει την περιγραφή του βήματος.
2. Στην μέση αριστερά περιέχει την εμφάνιση της πράξης σε κάθετη μορφή.
3. Στην μέση δεξιά είναι μια στήλη που εμφανίζονται τα δανεικά ή τα κρατούμενα από το κάθε βήμα.
4. Στο κάτω μέρος έχει 2 κουμπιά ένα που κλείνει το τρέχων παράθυρο και ένα που σε πάει στο επόμενο βήμα.

3.2 Το παράθυρο επιλογής για προσομοίωση η μετάβαση στην επόμενη πράξη.



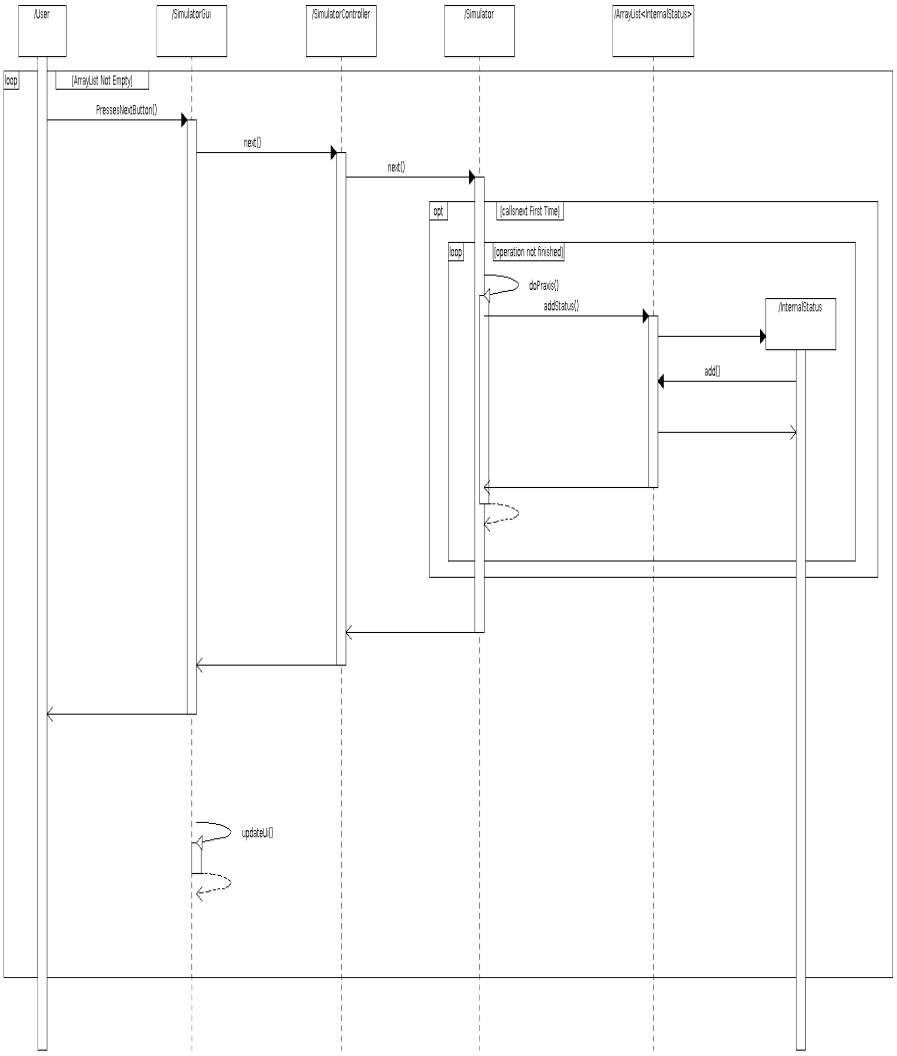
Εικόνα 6: Πάράθυρο επιλογής προσομοιωτή

Η προσομοίωση ξεκινά όταν κάνει κλικ στο κουμπί "Προσομοίωση πράξης" στο παράθυρο με τίτλο "Τέλος Προσπαθειών" που βλέπετε στην εικόνα 8.

4 Αλληλεπίδραση Κλάσεων

Προϋποθέσεις:

- Ο Χρήστης έχει ήδη κάνει λάθος 3 φορές μια πράξη και έχει επιλέξει προσομοίωση.
- Εδώ περιγράφουμε μια γενική αλληλεπίδραση μεταξύ των κλάσεων δεν θα περιγράψουμε πως αλληλεπιδρούν σε κάθε είδος πράξης.



Εικόνα 7: Διάγραμμα ροής κατά την διάρκεια μιας προσομοίωσης ενός παιχνιδιού

Παρατηρήσεις:

- Όταν καλείτε για πρώτη φορά η μέθοδος next() στον Προσομοιωτή τότε καλεί την μέθοδο doPraxis() που αναλόγως το πως την έχουμε υλοποιήσει

μέσω κληρονομικότητας γεμίζει καταλλήλως μια ArrayList από εσωτερικές καταστάσεις.

- Σε κάθε βήμα πράξεις μπορεί η ArrayList να γεμίζει με παραπάνω από μια καταστάσεις. Έτσι μπορώ να δείχνω όλες τιε ενέργειες αναλυτικά που γίνεται σε κάθε βήμα.

- Εφόσον γεμίζει μετά η next() στον προσομοιωτή απλά επιστρέφει μια εσωτερική κατάσταση που με την σειρά της ανανεώνει έμμεσα το γραφικό περιβάλλον. Μέσω της μεθόδου updateUI().

5 Επίλογος

Σε αυτήν την ενότητα είδαμε για τον προσομοιωτή των βασικών αριθμητικών πράξεων. Αποτελεί ιδιαίτερο χαρακτηριστικό που επιτρέπει στον μαθητή να βλέπει πως γίνεται βήμα βήμα μια βασική αριθμητική πράξη που έχει κάνει λάθος. Είδαμε πως αλληλεπιδρούν οι κλάσεις μεταξύ τους μέσω UML διαγράμματος ροής.

4 ο ΚΕΦΑΛΑΙΟ

Ο ΜΗΧΑΝΙΣΜΟΣ ΑΝΟΙΓΜΑΤΟΣ ΕΝΟΣ ΠΑΙΧΝΙΔΙΟΥ ΑΠΟ ΤΟ WEB

1 Εισαγωγή

Ένα ακόμη ιδιαίτερο χαρακτηριστικό του guma είναι ότι μπορεί να ανοίγει ένα αρχείο με κατάληξη .guma που είναι αποθηκευμένο στο web. Έτσι ένας δάσκαλος μπορεί μέσω υπηρεσίες cloud πχ. Dropbox ή και Ubuntu One. Να ανεβάζει ένα αρχείο και ο μαθητής να το ανοίγει από το σπίτι του σαν άσκηση.

Στην ουσία αυτό που κάνει είναι να αποθηκεύει στα προσωρινά αρχεία του συστήματος το απομακρυσμένο παιχνίδι και μετά να το ανοίγει και να το φορτώνει.

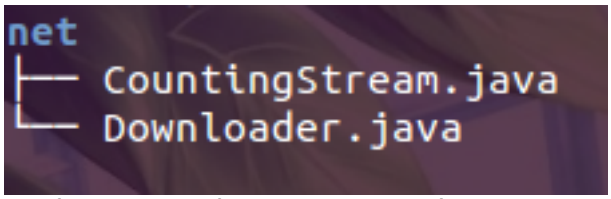
2 Εξωτερικές βιβλιοθήκες

2.1.1 Γενικά

Για τον σκοπό αυτό αξιοποιήθηκε η βιβλιοθήκη Apache Commons IO (http://commons.apache.org/proper/commons-io/) και δημιουργήθηκε ανεξάρτητα η βιβλιοθήκη net (https://github.com/pc-magas/net) που είναι μια βιβλιοθήκη

ελευθέρου λογισμικού (GNU GPL v3) που σαν σκοπό έχει να κατεβάζει αρχεία από το web.

2.1.2 Κλάσεις Βιβλιοθήκης net



Εικόνα 8: Οι κλάσεις της βιλιοθήκης net

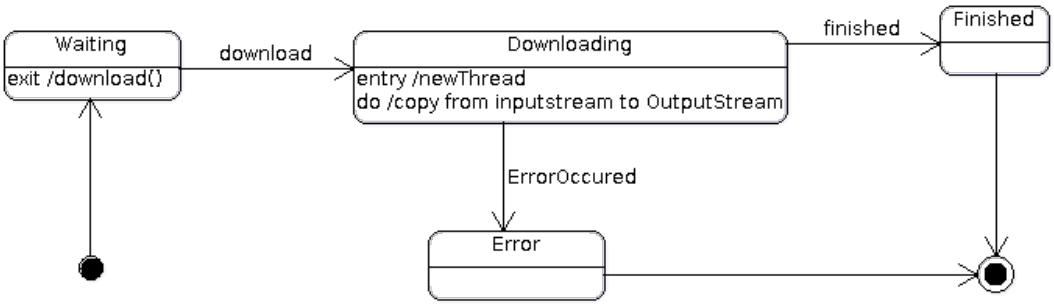
Η βιβλιοθήκη net αποτελείτε από 2 κλάσεις:

1. Η κλάση *CountingStream* που είναι μια κλάση που μετρά πόσο % έχει κατέβει το αρχείο.

2. Η κλάση Donwloader που είναι η κλάση υπεύθυνη να κατεβάζει το αρχείο.

3 Διαγράμματα Καταστάσεων

3.1.1 Κατάστασεις Κλάσης net.Downloader (από την Εξωτερική Βιβλιοθήκη)

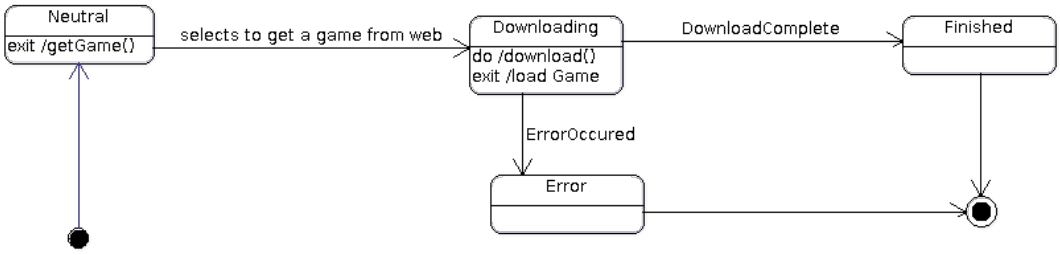


Εικόνα 9: Διάγραμμα καταστάσεων UML για την κλάση net.Downloader (ανεξάρτητη βιβλιοθήκη)

Παρατηρήσεις:

- Από την κατάσταση Waiting μόνο όταν ο χρήστης επιλέγει να κατεβάσει ένα αρχείο από το URL μεταβαίνει στην κατάσταση Downloading.
- Στην κατάσταση Downloading δημιουργεί ένα Thread που εκτελεί το download του αρχείου. Που στην ουσία είναι 2 Java Streams ένα InputStream που διαβάζει από το αρχείο από το web και στην μετά το αντιγράφω σε ένα OutputStream σαν να αντιγράφω ένα αρχείο. (Εδώ βοηθάει και η βιβλιοθήκη commons io από το ίδρυμα Apache).
- Αν δεν έχει προκύψει πρόβλημα μεταβαίνει στην κατάσταση Finished ειδάλλως στην κατάσταση Error.

3.1.2 Καταστάσεις κλάσης WebGameLoader (από το πακέτο guma.net)

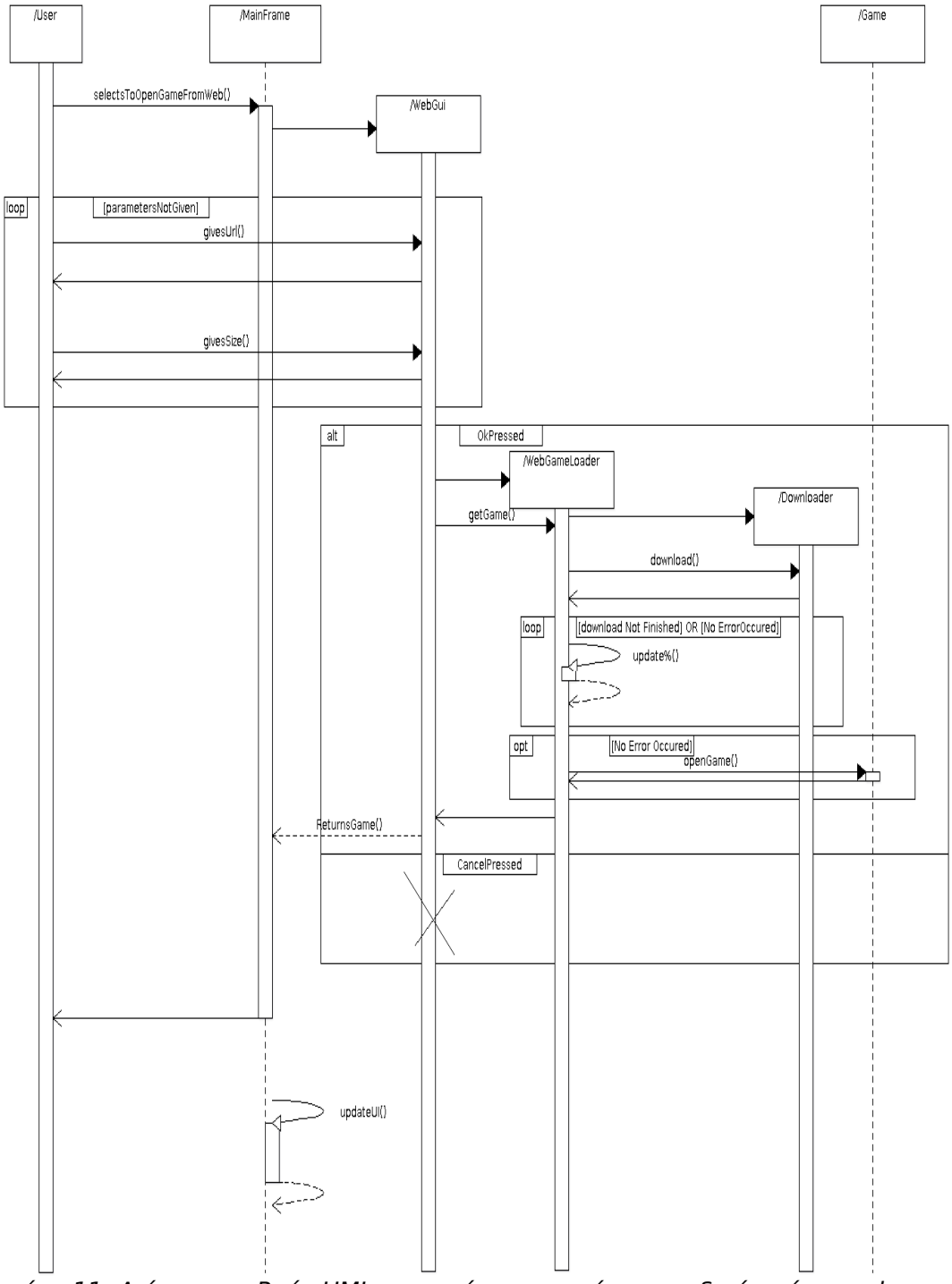


Εικόνα 10: Διάγραμμα καταστάσεων UML του guma.net.WebGameLoader

Παρατηρήσεις:

- Κατά το download κατεβάζει το αρχείο και μετά το ανοίγει.
- Αν δεν έχει προκύψει πρόβλημα μεταβαίνει στην κατάσταση Finished ειδάλλως στην κατάσταση Error.

4 Αλληλεπίδραση κλάσεων



Εικόνα 11: Διάγραμμα Ροής UML για το άνοιγμα ενός παιχνιδιού από το web

Παρατηρήσεις:

- Τα giveURL και τα giveSize είναι ενέργειες χρήστη, που δίνει αντίστοιχα το URL του αρχείου και το Μέγεθος .
- Δεν μοντελοποιώ στο διάγραμμα τυχών σφάλματα της διαδικασίας openGame().
- Δεν υπάρχει μέθοδος SelectToOpenGameFromWeb() είναι επίσης ενέργεια χρήστη.

5 Επιλογος

Σε αυτό το κεφάλαιο σας έδειξα ένα ακόμη ιδιαίτερο χαρακτηριστικό του guma, να ανοίγει αρχεία από το web. Σας εξήγησα τι Εξωτερικές Βιβλιοθήκες χρειάζονται όπως ακόμη και διαγράμματα UML που περιγράφουν της καταστάσεις τόσο των custom βιβλιοθηκών που έγιναν όσο και την Αλληλεπίδραση των μεταξύ κλάσεων για να γίνει το κατέβασμα και άνοιγμα ενός αρχείου από το web.

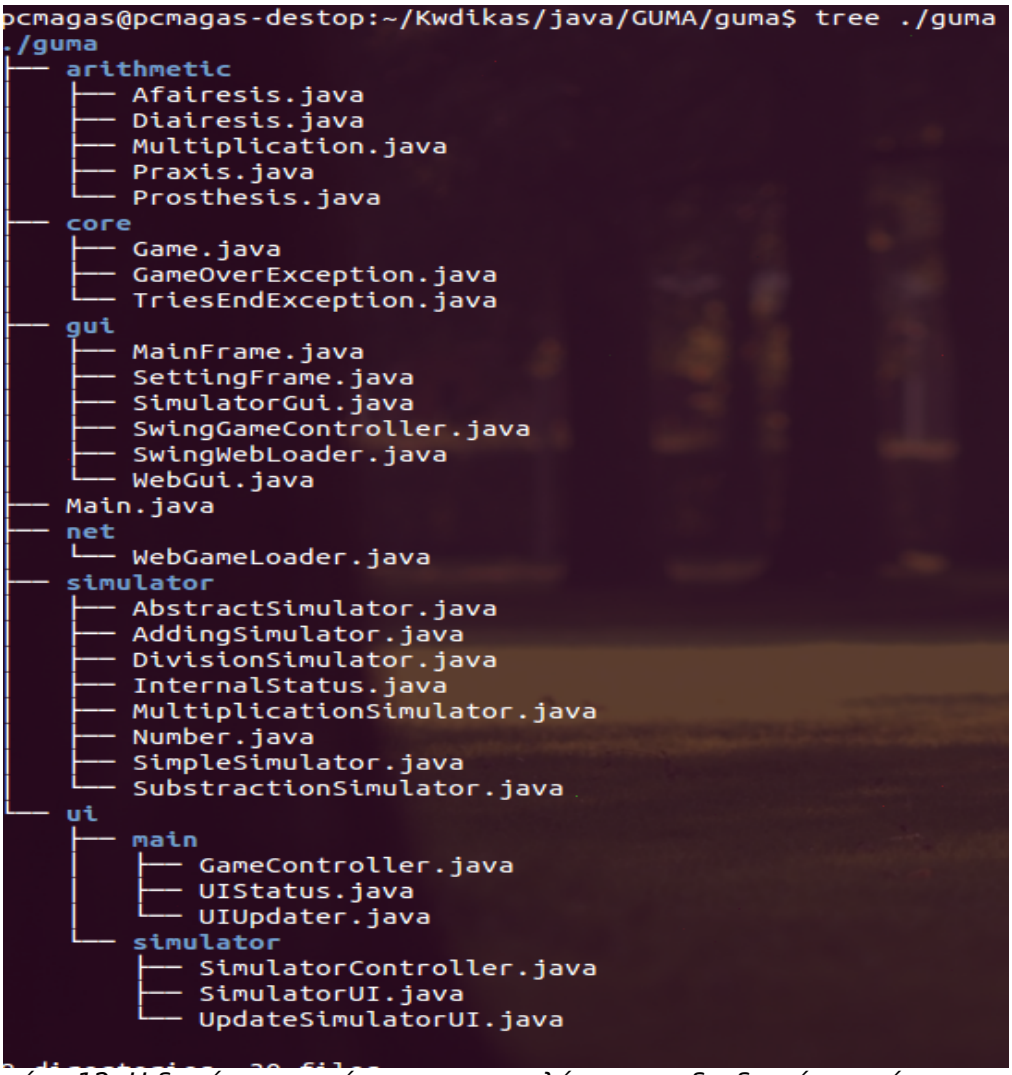
5 ΚΕΦΑΛΑΙΟ

ΟΙ ΚΛΑΣΕΙΣ ΚΑΙ ΤΑ ΠΑΚΕΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

1 Εισαγωγή

Για το τελευταίο κεφάλαιο θα σας αναφέρω πως δομείτε ο κώδικας στο σύνολό του. Δηλαδή από ποιες κλάσεις αποτελείτε και από ποια πακέτα. Ακόμη θα σας περιγράψω τι κάνει κάθε κλάση του guma καθώς και ποιες κλάσεις αντιστοιχούν στο καθένα από τα 3 λογικά μέρη της εφαρμογής. Τέλος θα δούμε και τις εξαρτήσεις μεταξύ των κλάσεων μέσα από ένα διάγραμμα Κλάσεων UML.

2 Οι κλάσεις και τα πακέτα της εφαρμογής



Εικόνα 12: Η δομή των πακέτων και των κλάσεων σε δενδρική μορφή

2.1 Τα πακέτα του guma

Η εφαρμογή αποτελείτε στην παρούσα φάση από ένα πακέτο το guma που περιέχει τα εξής υποπακέτα:

- Το πακέτο *arithmetic* περιέχει όλες τις κλάσεις που σχετίζονται με τις βασικές πράξεις που αξιοποιεί το παιχνίδι. **Δεν** περιέχει τις πλάσεις που συσχετίζονται με την προσομοίωση.

- Το πακέτο *core* περιέχει ότι κλάσεις συσχετίζονται με το παιχνίδι. Δηλαδή με λίγα λόγια αποτελεί τον πυρήνα της εφαρμογής αποτελείτε από τις βασικές κλάσεις του παιχνιδιού καθώς και κάποιες ειδικές κλάσεις που χρησιμοποιούνται σαν Exceptions.

- Το πακέτο *simulator* περιέχει κλάσεις που συσχετίζονται με την προσομοίωση των βασικών πράξεων.

- Το πακέτο *ui* περιέχει πακέτα, κλάσεις και διεπαφές που αποτελούν διασύνδεση μεταξύ εμφάνισης (πακέτο *gui* και πυρίνα) περιέχει 2 υποπακέτα:

    - Το υποπακέτο *main* που έχει κλάσεις και διεπαφές που λειτουργούν σαν διασύνδεση για την κυρίως εφαρμογή με την εμφάνηση.

    - Το υποπακέτο *simulator* που περιέχουν κλάσεις και διεπαφές που λειτουργούν σαν διασύνδεση για τον προσομοιωτή με την εμφάνιση.

- Το υποπακέτο *gui* έχει όλες τις κλάσεις που συσχετίζονται με το γραφικό περιβάλλον.

Όλα τα πακέτα υπάρχουν υπό ένα πακέτο *guma* που είναι και το κυρίως πακέτο της εφαρμογής

2.2 Οι κλάσεις που περιέχει το guma

Το guma περιέχει τις εξής κλάσεις σύμφωνα με την εικόνα 1:

- Υποπακέτο *arithmetic*:
    - Κλάση *Praxis*: είναι μαι αφηρημένη κλάση που αναπαριστά μια αριθμιτική πράξη 2 αριθμών.
    - Κλάσεις *Prosthesis*, *Afairesis*, *Multiplication*, *Diairesis*: Κλάσεις ποτ αναπαριστούν την πράξη τηε πρόσθεσης, της αφαίρεσης του πολλαπλασιασμού και της διαίρεσης αντιστοίχως.
- Υποπακέτο *core* περιέχει:
    - Κλάση *Game*: Κλάση που προσομοιώνει το κυρίως πρόγραμμα – παιχνίδι.
    - Κλάση *GameOverException*: Κλάση που σηματοδοτεί το τέλος ενός παιχνιδιού και η κλάση *Game* την "πετά" (throws) σαν Exception.
    - Κλάση *TriesEndException*: Έχει ίδια φιλοσοφία και ισχύουν ακριβώς τα ίδια όπως και με την *GameOverException,* με την μόνη διαφορά ότι αυτή "πετάγεται" όταν ο χρήστης έχει τελειώσει τον αριθμό των διαθέσιμων προσπαθειών του.
- Υποπακέτο *gui*:
    - Κλάση *MainFrame*: Μοντελοποιεί το κυρίως παράθυρο της εφαρμογής. Είναι η πρώτη ουσιαστικά κλάση που καλείται από την Κλάση *Main*.
    - Κλάση *SettingFrame*: Δημιουργεί το παράθυρο ρυθμίσεων του παιχνιδιού και καλείτε όταν ο χρήστης δημιουργεί ένα νέο παιχνίδι.
    - Κλάση *WebGui*: Είναι η κλάση που μοντελοποιεί το παράθυρο φόρτωσης παιχνιδιού μέσω από το web.
    - Κλάση *SimulatorGui*: Αποτελεί την υλοποίηση του παραθύρου του προσομοιωτή πράξεων.

- Κλάση *SwingGameController*: Επεκτείνει την κλάση *GameCotrnoller* από το υποπακέτο *guma.ui.main*, δίνοντάς του την δυνατότητα να εμφανίζει τα κατάλληλα παράθυρα εισόσου υπολοίπου καθώς και μηνυμάτων - σφαλμάτων.

- Υποπακέτο *net*:

  - Κλάση *WebGameLoader*: Μοντελοποιεί την κλάση που κατεβάζει το παιχνίδι και το ανοίγει. Ουσιαστικά είναι ο "φορτωτής" του παιχνιδιού από το web.

- Υποπακέτο *simulator*:

  - Κλάση AbstractSimulator: Όπως υποδηλώνει το όνομα αποτελεί μια αφηρημένη μοντελοποίηση ενός προσομοιωτή αριθμητικών πράξεων με 2 αριθμούς.

  - Κλάση SimpleSimulator: Αποτελεί λιγότερο αφηρημένη υλοποίηση προσομοιωτή που βοηθάει καλύτερα να αναπτυχθεί προσομοιωτής για τις πράξεις πρόσθεσης και αφαίρεσης.

  - Κλάση *Number*: Προσομοιώνει έναν αριθμό που συμμετέχει σε μια πράξη. Ουσιαστικά αποθηκεύει έναν αριθμό είτε σαν ένα πίνακα που περιέχει διαχωρισμένα τα ψηφία του είτε με την κανονική του (χωρίς διαχωρισμένη σε ψηφία τιμή) καθώς προσφέρει και ποιο ευκολία πρόσβαση στο να αλλάζεις μια τιμή ενός ψηφίου.

  - Κλάσεις *AddingSimulator*, *SubstractionSimulator*, *MultiplicationSimulator*, *DivisionSimulator*: Αποτελούν κλάσεις που μοντελοποιούν την προσομοίωση των πράξεων πρόσθεσης, αφαίρεσης, πολλαπλασιασμού και διαίρεσης η κάθε μια από αυτές.

- Υποπακέτο *ui*:

  - Υποπακέτο *main*:

- Κλάση *GameController*: Μοντελοποιεί την λογική τις επικοινωνίας παιχνιδιού με τον γραφικό ή εκάστοτε περιβάλλον.

- Κλάση *UIStatus*: Κλάση που περιγράφει την διεπαφή του χρήστη με το κυρίως μέρος της εφαρμογής, και αποθηκεύει τις τιμές που θα έχει κάποια βασικά πεδία της διεπαφής με τον χρήστη. (Η Διεπαφή του χρήστη είναι το γραφικό περιβάλλον αλλά θέλω να υπάρχει δυνατότητα να προσαρμόζεται σε οποιοδήποτε περιβάλλον).

- Διεπαφή *UIUpdater*: Είναι η διεπαφή (Interface) που αξιοποιήτε από το εκάστοτε γραφικό περιβάλλον.

◦ Υποπακέτο *simulator*:

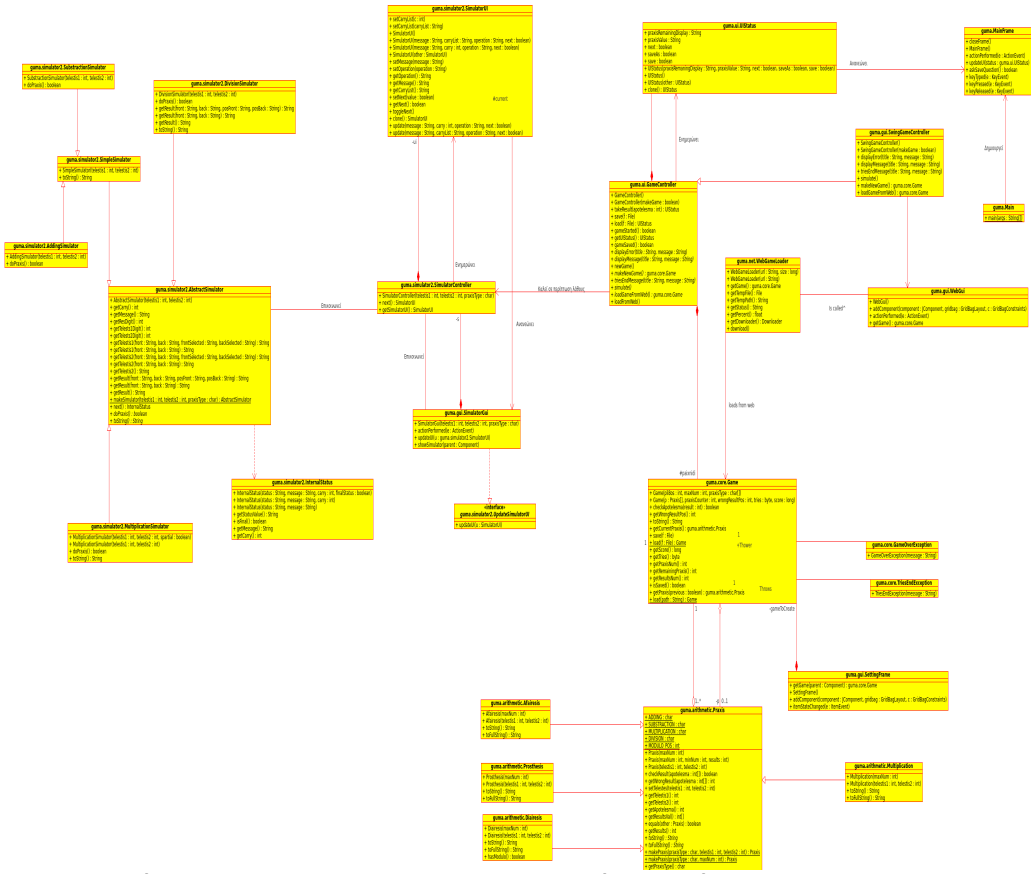
- Κλάση *SimulatorCotroller*: Μοντελοποιεί την Λογική επικοινωνίας του προσομοιωτή με το γραφικό περιβάλλον του.

- Κλάση *SimulatorUI*: Κλάση που περιγράφει την διεπαφή του χρήστη με το τον προσομοιωτή, και αποθηκεύει τις τιμές που θα έχει κάποια βασικά πεδία της διεπαφής με τον χρήστη. Όπως κάνει και η κλάση *UIUpdater* στην κυρίως εφαρμογή.

- Κλάση *UpdateSimulatorUI*: Διεπαφή που έχει σαν στόχο να ανανεώνει την διεπαφή χρήστη προσομοιωτή σύμφωνα τις τιμές που αποθηκεύει η κλάση *SimulatorUI.*

2.3 Διάγραμμα Κλάσεων:

3 Συσχετισμός πακέτων με τα εκάστοτε μέρη της εφαρμογής



Εικόνα 13: Οι Κλάσεις του Guma και οι συσχετισμοί μεταξύ τους

Η εφαρμογή έχει τις εξής κλάσεις σε κάθε μέρος:

1. Για την *κυρίως εφαρμογή*: Αποτελείτε από τα υποπακέτα guma.artithmetic, guma.core, guma.ui.main και τις κλάσεις που εμπεριέχονται σε αυτά καθώς και τις κλάσεις από το υποπακέτο guma.gui MainFrame, SettingFrame, SwingGameController, WebGui.

2. Για τον *προσομοιωτή πράξεων*: Αποτελείτε από τα υποπακέτα guma.simulator, guma.ui.simulator και τις κλάσεις που εμπεριέχονται σε αυτά καθώς και την κλάση SimulatorGui από το υποπακέτο guma.gui.

3. Για τον *μηχανισμό φόρτωσης παιχνιδιού από το web*: Αποτελείτε από την κλάση WebGameLoader από το υποπακέτο guma.net καθώς και την κλάση

WebGui από το υποπακέτο guma.gui.

Αυτός ο διαχωρισμός γίνεται προκειμένου να μελετηθεί ποιο εύκολα η εφαρμογή.

4 Επίλογος

Σας έδειξα το περιβάλλον της εφαρμογής και σας περιέγραψα περιληπτικά τι κάνει καθώς και γιατί επέλεξα την java σαν γλώσσα προγραμματισμού. Ακόμη σας έδειξα την βασική δομή της εφαρμογής και εστιάσαμε στην δομή των κλάσεων και των υποπακέτων αλλά και το πως συσχετίζονται μεταξύ τους οι κλάσεις. Τέλος σας περιέγραψα τον λογικό διαχωρισμό της εφαρμογής σε μέρη προκειμένου να διευκολυνθεί η μελέτη και η κατανόηση της εφαρμογής

6 ο ΚΕΦΑΛΑΙΟ

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

1 Εισαγωγή

Το guma είναι ένα project που δεν έχει σαν στόχο να ολοκληρωθεί στα πλαίσια μιας πτυχιακής αλλά έχω σαν στόχο να το συνεχίσω σαν ένα project που θα εξελίσσεται στα πλαίσια μιας κοινότητας.

2 Μελλοντική στόχοι του project.

Κατ αρχάς ένας από του στόχους είναι η δημιουργία μιας κοινότητας που θα παρέχει υποστήριξη σε χρήστες αλλά και θα αναπτύσει, θα δοκιμάζει και θα τεκμηριώνει το guma. Ακόμη θα περιέχει γονείς και μαθητές αλλά και δασκάλους που θα μας καθοδηγούν έτσι να γίνει μια πλήρης εκπαιδευτική πλατφόρμα. Ένας ακόμη στόχος είναι να συμμετέχουν στις δοκιμές και μαθητές Δημοτικού εξάλλου αυτό είναι το μεγαλύτερο μέρος του target group.

Πέρα από αυτό ένας από τους σκοπούς είναι να τρέχει πέρα από desktop εφαρμογή και σαν web μέσω tomcat και μέσω html5 και javascript να μπορεί να τρέχει και σε

συσκευές με χαμηλή επεξεργαστική ισχύ, που συνήθως διαθέτουν τα ελληνικά σχολεία. Δε λειτουργικά όπως firefox OS (Mozilla Developenr Network) και το Ubuntu Touch (Ubuntu Developers) έχουν δυνατότητα να έχουν στο σαν εγκατεστημένη εφαρμογή εφαρμογές που είναι σε html5 και Javascript στο frontent και είναι πάνω στο web.

Πέρα από αυτό ένας άλλος στόχος είναι να υπάρχει και πλατφόρμα που θα μπορεί να "ανεβάζει" ο καθηγητής custom παιχνίδια και τεστ – διαγωνίσματα πάνω στο web έτσι ο μαθητής στο σπίτι η στην αίθουσα να μπορεί να τα τρέχει.

Τέλος ένας άλλος στόχος είναι να υπάρχει δυνατότητα καταγραφής προόδου έτσι η εφαρμογή να κρατά στατιστικά για τον μαθητή και να παράγει αναφορές για τον κάθε έναν μαθητή στο πόσο καλός είναι σε μια πράξη έτσι να μπορεί ο καθηγητής ή και ο γονέας να γνωρίζει που ο εκάστοτε μαθητής. Έτσι υπάρχει η δυνατότητα προσαρμογής της διδασκαλίας προκειμένου να καλυφθεί αυτή η αδυναμία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Βιβλίο:

   Ian Sommervile, Βασικές Αρχές τεχνολογίας Λογισμικού (2011), Αθήνα:Κλειδάριθμος

2. Σημειώσεις:

   Ευάγγελος Σπύρου, Εργαστήριο Τεχνολογίας Λογισμικού, 2011-2012

3. Ιστότοπος:

   Wikipedia:http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

4. Ιστότοπος:

   MSDN: http://msdn.microsoft.com/en-us/library/ff649643.aspx

5. Ιστότοπος:

   Wikipedia: http://en.wikipedia.org/wiki/Richard_Stallman

6. Ιστότοπος:

   Wikipedia: https://en.wikipedia.org/wiki/Free_software

7. Ιστότοπος:

   GNU Project: http://www.gnu.org

8. Ιστότοπος:

   FSF: http://www.gnu.org/philosophy/free-sw.html

9. Ιστότοπος:

Mozilla Developer Network: https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Application_development

10. Ιστότοπος:

    Ubuntu Developers:http://developer.ubuntu.com/get-started/

ΠΑΡΑΡΤΗΜΑΤΑ

ΤΟ ΜΟΝΤΕΛΟ MVC

1 Γενικά:

Το μοντέλο MVC είναι η αρχιτεκτονική που έχει το εξής χαρακτηριστικό: Να διαχωρίζει την εμφάνιση από την από την διεπαφή του χρήστη με την εφαρμογή. Το μοντέλο αυτό είναι διαδεδομένο στις web εφαρμογές ακόμη και σε γνωστά CMS όπως το Joomla.

2 Γιατί το χρειαζόμαστε:

Σύμφωνα με το msdn[4] η αρχιτεκτονική mvc μπορεί να δώσει λύση στα εξής προβλήματα :

- Η εμφάνιση της εφαρμογής αλλάζει ποιο συχνά από ότι η λογική της έτσι θέλουμε έναν τρόπο που να μπορούμε να ανανεώνουμε εύκολα το πως φαίνεται σε σχέση με την υπόλοιπη εφαρμογή.
- Πολλές φορές η εφαρμογή εμφανίζει τα ίδια δεδομένα με διαφορετικούς τρόπους και πολλές φορές από διαφορετικές συκσευές. Πχ. Μπορούμε έναν πίνακα να το θέλουμε σε Html για τον Browser αλλά και σαν PDF για να μπορούμε να το εκτυπώσουμε.
- Πολλές φορές ο σχεδιασμός του περιβάλοντος μιας εφαρμογής απαιτεί έξτρα δυνατότητες από τον προγραμματισμό πχ. Στο web design.

- Ο σχεδιασμός της εμφάνισης είναι ποιο χρονοβόρος από την ανάπτυξη της λογικής της εφαρμογής.

3 Δομή της Αρχιτεκτονικής:

Η αρχιτεκτονική αποτελείτε από 3 μέρη:

1. ***View:*** Αποτελεί την εμφάνιση της εφαρμογής. Με απλά λόγια την διεπαφή του Χρήστη με την εφαρμογή. Ανανεώνεται σύμφωνα από την Wikipedia[3] από το μοντέλο.

2. ***Model:*** Στο μοντέλο αναπτύσσεται όλη η αρχιτεκτονική του εκάστοτε συστήματος λαμβάνει μηνύματα από τον ***controller*** για τυχόν αλλαγές στην κατάστασή του . Ακόμη στέλνει μηνύματα στον ***cotnroller*** για να ανανεώσει στην συνέχεια το view,

3. ***Controller:*** Είναι ο υπεύθυνος επικοινωνίας μεταξύ ***model*** και ***view*** λαμβάνει και στέλνει μηνύματα στο ***model*** ανάλογα από τα ερεθίσματα που λαμβάνει από το view.

Η ΦΙΛΟΣΟΦΙΑ ΤΟΥ ΕΛΕΥΘΕΡΟΥ ΛΟΓΙΣΜΙΚΟΥ

1 Εισαγωγή:

Είναι η φιλοσοφία διαμοιρασμού και ανάπτυξης λογισμικού το οποίο σέβεται την ελευθερία του χρήστη και του επιτρέπει να έχει τον πλήρη έλεγχο πάνω σε αυτό. Να ξεκαθαρίσουμε όταν εννοούμε ελευθερία δεν σημαίνει ότι δεν έχει χρηματικό κόστος όπως αναφέρει και ο Richard Stallman[5] *"Free software is a matter of*

liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer"[6].

Το κίνημα αυτό ξεκίνησε από τον Richard Stallman και με το project GNU αντιδρώντας στο ότι την δεκαετία του '80 άρχισαν να αναπτύσσονται λογισμικά και προγράμματα που δεν επέτρεπαν τον ελεύθερο διαμοιρασμό κώδικα και λογισμικού μεταξύ των χρηστών η ακόμα και την τροποποίησή αυτού.

2 Η φιλοσοφία:

Η φιλοσοφία αποτελείτε από τα εξής (FSF):

1. Την ελευθερία να τρέχεις το πρόγραμμα για οποιοδήποτε σκοπό.
2. Την ελευθερία να μελετάς το πως το πρόγραμμα λειτουργεί και την δυνατότητα να το τροποποιείς σύμφωνα με τις ανάγκες σου. Άρα απαραίτητο είναι να διανέμεται και ο κώδικας μαζί.
3. Την ελευθερία να το μοιράζεις και σε τρίτους.
4. Την ελευθερία να μοιράζεις τροποποιημένη έκδοση αυτού. Έτσι να δίνεις σε όλη την κοινότητα την δυνατότητα να επωφεληθούν από αυτό.

3 Άδειες:

Υπάρχουν πολλές άδειες που μπορούν να σου δώσουν αυτήν την δυνατότητα. Η ποιο γνωστή είναι η GNU General Public Licence. Παρόλα αυτά υπάρχουν και άλλες άδειες που έχουν αυτήν την φιλοσοφία. Μερικές είναι ή Apache Commons 2, GNU LGPL, GNU AGPL, X11 Licence κλπ κλπ.