



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021



Ξηρουδακης Γεωργιος.

AM: csd5067

10/1/2023

Εισαγωγή

Σκοπος αυτου του project ειναι η διασκευη του επιτραπεζιου παιχνιδου stratego σε ηλεκτρονικη μορφη.

Περιεχόμενα

<u>1.Εισαγωγή.....</u>	<u>2</u>
<u>2.Η Σχεδίαση και οι Κλάσεις του Πακέτου Model</u>	<u>3</u>
<u>3.Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller</u>	<u>3</u>
<u>4.Η Σχεδίαση και οι Κλάσεις του Πακέτου View</u>	<u>3</u>
<u>5.Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....</u>	<u>3</u>
<u>6.Λειτουργικότητα (Β Φάση).....</u>	<u>4</u>
<u>7.Συμπεράσματα.....</u>	<u>4</u>

1. Εισαγωγή

Το μοντελο που χρησιμοποιειτε για την υπολοιψη αυτην της εργασιας ειναι το MVC (Model View Controller) .Περιληπτικα ενα MVC μοντελο αποτελειτε απο τρεια διαφορετικα πακετα. 1) το model οπου ειναι το backend και χηριζεται την διαχηρηση των δεδομενον και τα αποτελεσματα των διενεργειων. 2) Το view οπου ειναι τελειως ανεξαρτητο απο το model και παραγει-ανανεωνωει ολα τα γραφηκα την εφαρμογης. Τελος 3) Ο controller οπου ουσιαστικα εννοει το model με το view, αφου ο controller για καθε ενεργεια που πρεπει να πραγματοποιηθει περναι τα δεδομενα και καλει τις καταληλες συναρτησεις απο το model και αφου απο αυτο παρει τα αποτελεσματα τα ανανεωνει καταληλα το view

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

2.1 piece

Συγουρα απο της ποιο συμαστικες κλασεις του model ειναι το piece και οι κλασεις που την κληρονομουν. Η piece εχει τα εξεις πεδια:

private Position position την θεση του κομματιου.

private Image frontImage την εικονα του κομματιου.

private Team team την ομάδα στην οποία ανηκει το κομματι

private boolean isCaptured αν το κομματι είναι ενεργο ή εχμαλοτισμενο.

Συνατησεις:

setters and getters για τα παραπάνω πεδια και constuctor

Movable pieces

Αυτη είναι επίσης μια abstract class που κάνει extend την piece και αναφερετε στα κομματια όπου μπορούν να κουνηθούν κατά την διάρκεια του παιχνιδιού(όλα τα κομματια εκτός από τις σημαίες και τις βομβές.)

έχει παραπάνω το πεδίο int rank, και boolean hasrevived αυτά αποφούν μόνο τα movable pieces

το rank χρησιμοποιητε στις μάχες ενώ το hasrevived 'θύματα' αν το πιόνι έχει κερδίσει κάποιο rival καθώς το κάθε πιόνι δικαιούται μόνο ένα.

Από επιπλέον συνατησεις έχουμε setters και getters για τα επιπλέον παιδιά και constructor .

Επίσης μια συνάρτηση όπου επιστεψει το αν μια κινήσει που προκειται να κάνει το συγκεκριμένο πιόνι προκειται για επίθεση ή απλά κινήσει αυτό το κάνει χρησιμοποιώντας το boardcopy που θα εξηγήσουμε παρακάτω.

Επίσης έχει την συνατηση determinewinner όπου αποφασίζει ποιο από δύο πιόνια νικάει στην μεταξύ τους μάχη. Λαμβάνει υποψέιν τις ιδιικές ικανότητες κάποιων πιόνων και τα rank τους

ακόμα η συνάρτηση move το κάνει τις κατάλληλες αλλαγές για να μετακινηθεί ένα πιόνι σε μια διαφορετική θέση

Θελος η attack κάνει τις κατάλληλες αλλαγές για να γίνει μια επίθεση ενός πιονιού σε ένα αντιπαλο

Imovable pieces

αντίστοιχα τα κομματια που δεν κουνιούνται δημιουργούνται από μια άλλη abstract class Imovable piece. Παρόλο που δεν έχει επιπλέον πεδια ή συνατησεις και πάλι υπάρχει πλεονεχτιμα να τα ξεχωρίσουμε καθώς μπορούμε με ένα instanceof να δούμε αν προκειται για τέτοιο κομματι.

Spesial piece

το ίδιο με τα imovable pieces ισχύει και εδώ.

Ολα τα ειδη των κομματιων πχ σημαια δρακος νανος κανουν extent μια απο τις παραπανω ubstract classes.

2.2

Board

το Board στην ουσηα ειναι ενα διωδιαστατος πινακας οπου αναπασα στιγμη αντιπροσωπευει το τι υπαρχει στο ταβλο του παιχνιδιου. Καθε συναστησει που κανει καποια αλλαγη στο ταμπλο αυτο το boardcopy περνει τις πληροφοριες που χρηαζετε και μετα αννανεωνει οτι αλλαζει.

Ο πινακας που περιεχει ειναι τυπου boardmembers οπου ειναι ουσιαστικα ενα structure που αντιπροσωπευει το καθε τετραγονο του ταμπλο. Εχει πεδια για το αν ειναι ligal αν μπορει καπποιο ποιονι να μετακινηθει εκει, και το ποιο πιονι εχει αυτην την στιγμη null αν ειναι αδειο.

Position

Λογο του οτι το παιχνιδι διαδραματιζετε σε ταμπλο χειριζομαστε πολυ συνετεγμενες. Αρα θεωρησα σκοπημο να υπαρχει μια κλαση Myposition οπου θα εχει δυο πεδια χ,υ και θα μας βοηθαει στην διαχρηση τους. Εχει συνατησεις για το αν μποουμε να παμε σε αυτο το σημειο αν ειναι επιτρεπτο κτλπ

2.3

player

player ειναι μια κλαση που αντιπροσωπευει ενα πεχτη εκει πεδια οπως το παρατσουκλι του το σε ποια ομαδα ανικει , εναν πινακα με τα κομματα του το ποσες μαχες εχει νικησει και σε ποσες εχει συμμετεχει(για τον υπολογισμο του win perscetage του) το αν εχει νικησει το πεχνιδι και το ποσα revives εχει κανει(καθως δικαιουται μεχτι 2)

απο συνρτησεις εχει seters and getters για τα παραπανω παιδια καθως και για τον υπολογισμο του win percentage του.

Για το μενου που παρασιαζετε χρηαζομαστες πολλες πληροφοριες για τον παιχτι και τα πιονια του αρα εχουμε συνατησεις που υπολογιζουν ποσσα εχμαλοτισμενα ποιονια εχει, και ποσα εχμαλοτισμνα απο το καθε ειδος. Τελος εχει και μια συνατηση που υπολογιζει το αν εχει χασει απο ελιψη κινήσεων(αυτο το κανει

περνοντας απο ολα τα πιονια του παιχτι και βλεποντας αν εχουν κινήσεις.

2.4

Team

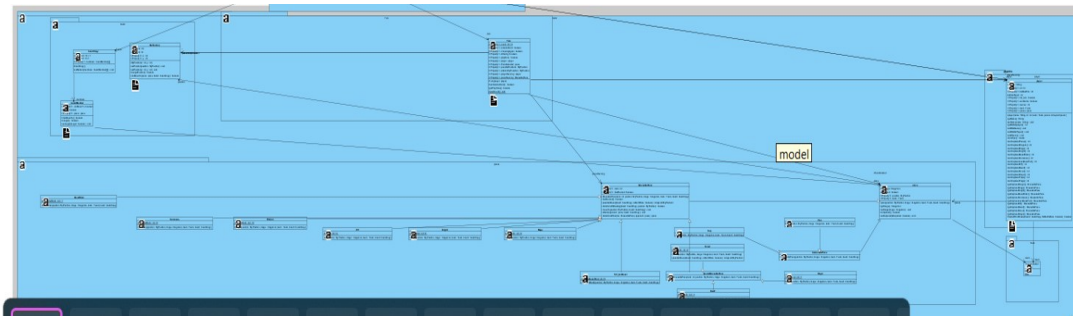
Επισεις πολλες φορες χρηαζεται να ξεουμε σε ποια ομαδα ανικει κατι αρα αντι να χρησιμοποιουμε ενα Int η ενα boolean γαι να ειναι ποιο ξεκαθαρο φτιαξαμε ενα enum Team οπου εχει δυο values FIRE, ICE.

2.5

Τελος το model εχει μια κλαση turn. Αυτη η κλαση ειναι παρα πολυ σθμαντικη για την συνεχεια του παιχνιδιου. Σε καθε γυρο ο πεχτις προτα διλεγει ενα ποιονι μετα διαλεγει το που θα μετακινηθει κτλπ αρα χρηαζομαστε μια δωμη γαι να ξεουμε σε ποιο βημα ειαμαστε. Σε καθε καινουρια σειρα δημιουργιεται ενα κενοθργιο στιγμαστηπο την κालης οπου αποθηκευονται ολα τα δεδομενα που χρηαζομαστε για αυτην πχ τον παιχτη το ποιονι που διαλεχει το αν επιτιθεται τα possible moves κτλπ. Επισεις εχει ενα στατικο πεδιο που μετραει το ποσα turns εχουν παιχτει οπου ανανεωνεται μεσο του constructor

εχτος απο constructor setters and getters εχει συνασρτηση για τον μυδενισμο του στατικου πεδιου σε περιπτωση που το πεχνιδι γινει restart

παρακατω φαινεται το διαγραμμα του model:



3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

3.1

Ο controller εχει πεδια για τα διαφορα uis(main ,revive ,win) για τους κανονες που ειναι ενεργει καθως και διαφορες στατικες τελικες για τα διαφορες τιμες πχ μεγαθος του ταμπλου που εμγανιζονατι τα πιονια καθε πεχτη αλλα και strings για τα paths των διφορων resorses για να ειναι ομοιομορφα και ποιο εφδιακριτα.

Ο constructor του condroller αρχικοποιει ολα τα uis τα action listenes των κουμπιων τους καθώς και την φωτογραφιες και τα δεδομενα στο ταμπλο και στο μενου.Δημιουργει δυο πεχτες και αναλογα με το αν ενεργοποιηθει ο κανοντας για μισο στρατο αρχικοποιει τα ποιονια τους τόσο στο ταμπο όσο και στους πινακες του καθε παιχτι τελος δημιουργει και αρχηκοποιη την πρωτη σειρα για τον παιχτη της φοτιας

η `intiPieces` είναι αυτη που δημιουργει τον πινακα με τα κοματια αναλογα την ομαδα και τους κανονες τα τοποθετει σε τυχρες θεσεις απο τις τρεις πρωτες σειρες που εχει ο καθε παιχτης (εχτος απο την σημεια που ειναι παντα στην πρωτη γραμη)

Για την δημιουργια την σπρας καθε φορα χρησιμοποιηται η συναρτηση newTurn (εχτος απο τν πρωτη φορα στο constructor) αυτη εχτος απο το να γτιαχει το κενουργιο στιγμηνιτηπο της σειρας με το καταληλο παιχτει κανει και αλλα πραγματα που πρπει να γινουν σε καθε σειρα πχ κριβει και ανοιγει τις καταλληλες καρτες, ανανεωνει τα μενου στα γραφικα και ελεχει αν ο παιχτης εχει κινηση να κανει.

Η `setTileImages` χρησιμοποιηται οποτε θελουμε να ανανεωσουμε τις background φωτογραφιες στο ταμπλο, ελενχει οπου δεν υπαρχει πινει απο το boardcopy και βαζει την αντιστιχη φωτοραγια.

Η `uptadeCaptures` χτησιμοποιηται για το παραστημα του μενου που διχνει τα κοματια που εχει εχμαλοτισει ο καθε πεχτης καλει αλλες συναρτησεις για το ποια φωτοραφια να βαλει που και το τι αριθμο να κανει attach σε καθε φωτοραφια.

Η `Revive` καλειτε οταν καποιος πεχτει κερδιζει revive δημιουργει ui με τις καταληλες φωτοραγιες και τους καταληλους αριθμους ωστοσο δεν γινεται εκει το revive καθώς ο παιχτης μπορεί να το αρνιθει και θα το κανει αργότερα αρα αυτο γινιταιο στον actionlistener των κουμπιων του revive ui

τελος η `playAndSaveSoundClip` ξεκαιναει ενα κενουργιο Clip thread και παιζει τον δωμενο ειχο επισεις υπαρχουν συνατησεις για να αποδηκευεσται αυτο το thread για να το σταματαμα κτλπ

Τελος υπαρχουν 4 actionListeners

πρωτος και μεγαλυετος ειναι για τα κουμπια του ταμπλου. Δουλειει πολυ στενα με την καλοη της σειρας καθώς ο τροπος που δουλειει ειναι σχεδον αναδρομικος αφου καλειτε πολλες φορες στην ιδια σειρα και αναλογα με το τι εχει αποδηκεθει στην σειρα κανεις της καταληλες ενεργειες. Αρχικα παντα βρξοκει το ποιο κουμπι του ταμπλου πατηθηκε. Αν ο πεχτει δεν εχει διαλεξει καρτα σημενει οτι τωρα διλεξε αυτην, και τελειωνει ενω αν ειχε διαλεξει σημενει οτι τωρα διαλεγει το που θελει να κουνισει το ποιονι του. Σε αθτην την περιπτωση βλεπει αν θλει να πετακινηθει η να κανει επιθεση και ενημερωνει καταληλα τόσο τα ui όσο και τα μελη του model πχ boardcopy. Ελενχει αν εχει νικησει καποιος αν κεδισε καποιο revive κτλπ. Τελος καλει την newturn για καινοθργια σειρα.

Δευερος ειναι γιαι το revive, οπως αναφερθηκε αυτος ειναι που οτος κανει το revive καθως ειναι εννομενος στα κουμπια που παταει ο παιχτης για να διαλεξει ποιο θελει να κανει revive. Ανανεωνει καταληλα το model και το view

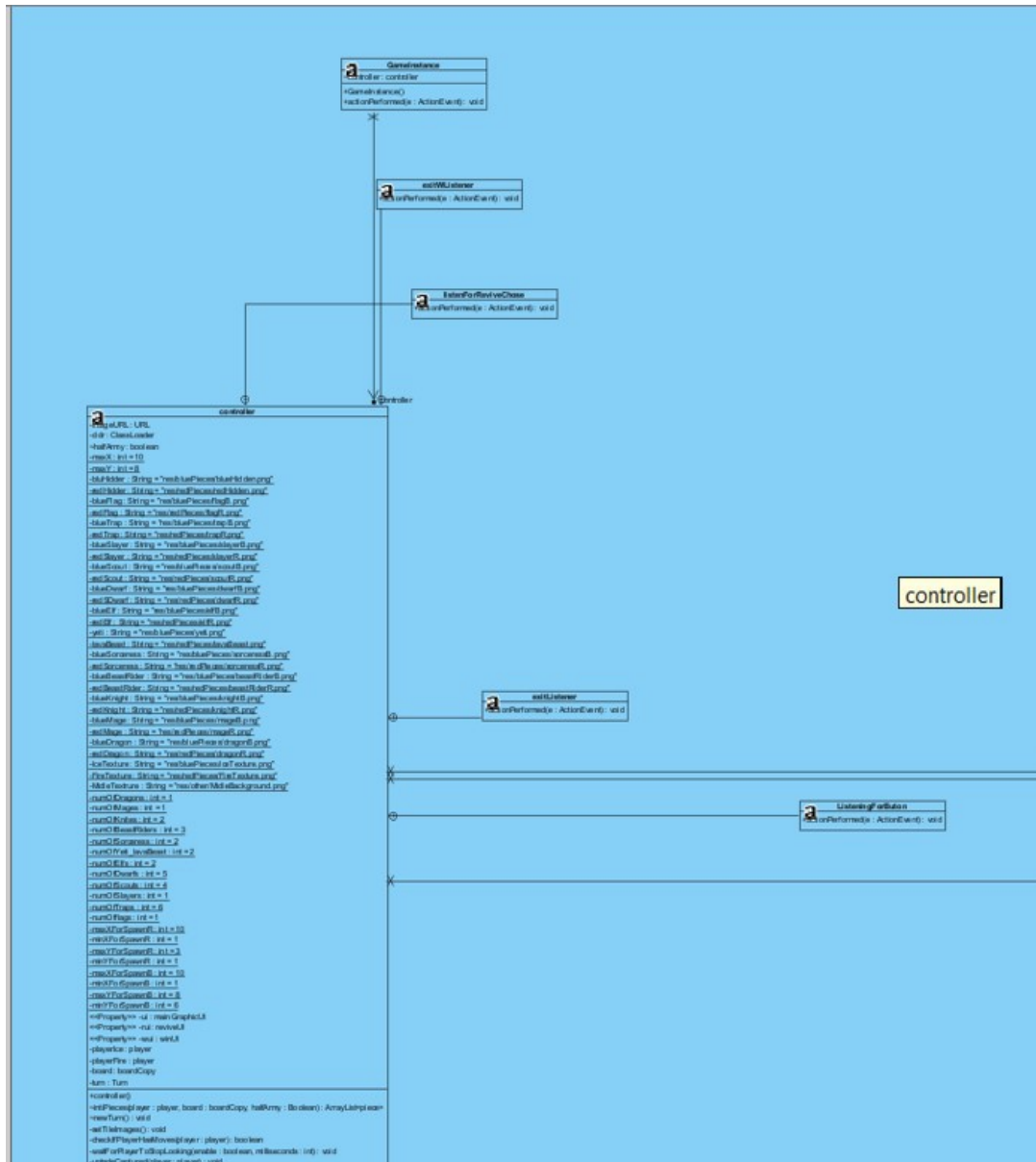
τριτων ο exiwindow χρησιμοποιητε για να κλεισουμε το παραθηρο του revive σε περιπτωση που δεν θελει ο πεχτης να κακνει revive

τελος ο closeWlistener καλειτε οταν θελουμε να τερματισουμε το παιχνιδι οταν καποιος νικησει η απο το mainui

3.2

η `GameInstance` ειναι μια καλη που υπολοιπει τον actionlistener και την ουσια δημιουργηθηκε για να υποστηριζεται το restart στον constructor δημιουργει κενουργιο controller ενω το action performed του κανει dispoce ολα του 'παλιου game instance και δημιουργει καινουργιο.

Παρακατω φαινεται το διαγράμματα του condroller:



4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

4.1 main ui

Το κυριο ui είναι το panel που έχει το board και το μενου. Το panel αυτο έχει boarder layout με το cender να είναι το board και δεξια το μενου.

Το board είναι ένα απλό grid 8 : 10 από κουμπιά όπου το κάθε κουμπι είναι ένα τετράγωνο του board

αντιστοιχα το μενου ειανι ενα panel με gird layout 3:1 για τα τρια διαφορετικα μελη(κανονες, στατιστικα, εχμαλοτισμενα) . Αυτα τα 3 υποπανελς χρησιμοποιουνε

κυριως bagGrid layouts για περισσοτερο ελενχο (περιεχου JLabels, Jsliders, Jbuttons αναλογα το τι χρειαζομαστε.)

οπως αναφεραμε το view δεν γνωρισει το model και γενικα την λειτουργια του απλα εμφανιζει ετοιμα δεδομενα που του δηνουμε. Αρα κατα συνεπεια εχει αρκετα απλες συναρτησεις .

Setters and getters για τα διαφορα labels κτλπ που υπαρχουν στο μενου(προσαρμοσμενα πχ λαμβανουν ενα int και αλαζουν το text να διχνει αυτο το Int)

Η `setImageInButton` βαζει την δωσμενη φωτογραφια στις δωσμενες συνεταγμενες του board . Αυτη η συναρτηση επισης χρησιμοποιει μια συναρτηση resize οπου αναλογα το δωσμενο μεγαθος του button κανει resize την εικονα.

Η `showPossibleMove` χρησιμοποιηται για να δειξουμε στον παιζητη που μπορεί να μετακινηθει με το να δημιουργουμε κοκκινο boarder αν ειναι επιθεση η πρασινο αν ειναι απλη μετακινηση. Αντιστηχα και η `stopshowingPossibleMove` για αν βγαλουμε το boarder

Η `displayErrorMessage` Χρησιμοποιηται για να δηξουμε μηνηματα στον παιχτη πχ οτι δεν ειναι σειρα του ή οτι δεν μπορεί να κανει αυτην την κινηση.

Τελος υπαρχουν συναρτησης που κανουν add actionlisteners σε διαφορα buttons (καλουνται απο τον κοντρολες που τους υλοποιει).

4.2 revive ui

αυτο το ui γινεται visible οταν καποιος πεχτεις κερδιζει ενα revive εχει και αυτο boarder layout και εχει σαν cender ενα grid με τις φωτογραφιες των ποιονων σε κουμιοια και ποσα απο το καθενα εχουν γινει captured τα μονα enabled κουμπια ειναι αυτα οπου ειναι τουλαχιστον ενα captured. Πανω ειναι ενα επεξηγηττικο label και κατω ενα κουμπι exit για αμα δεν θελει να κανει το revive

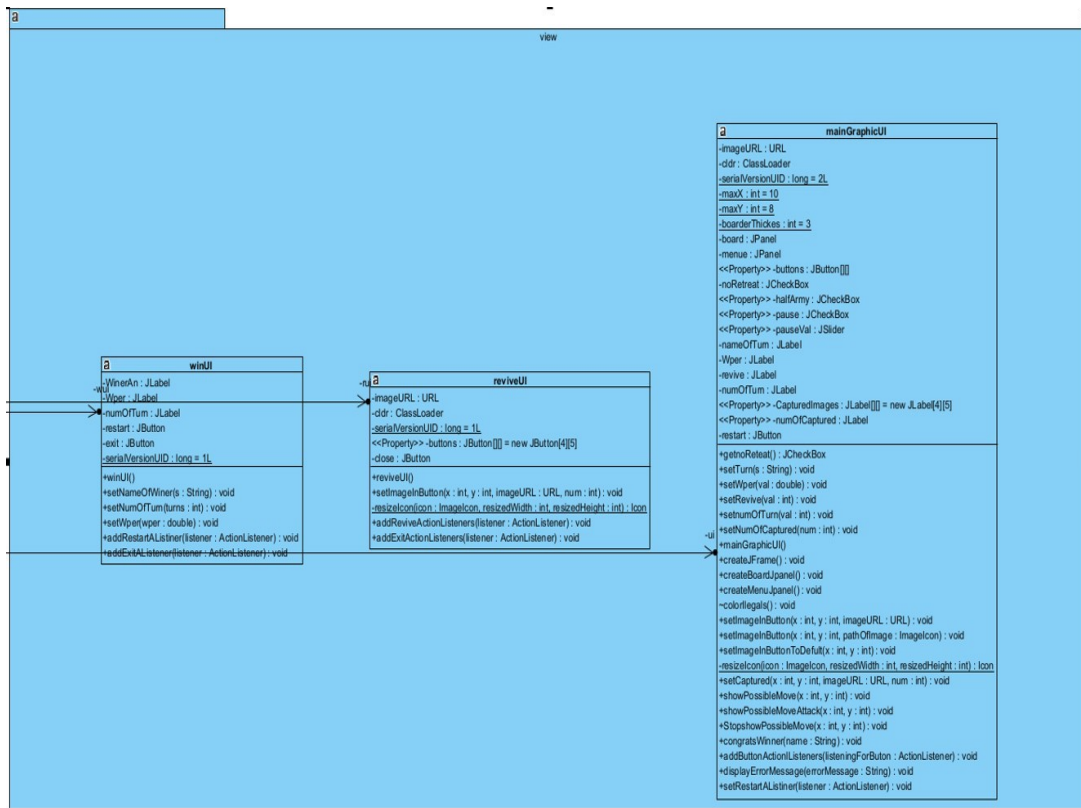
οι μονες συναρτησεις που εχει ειναι setters για τα διαφορα jlbels και buttons (φωτ και αριθμους), καθώς και συναρτησεις για adding actionListeners

4.3 win ui

αυτο το ui γινεται visible μονο οταν καποιος παιχτης νικησει. Εχει και αυτο boarder layout πανω εχει label που λεει ποιος κερδισε στο κεντρο labels για καποια στατιστικα και τελος bottom δυο κουμπια ενα για να κανει restart το παιχνιδι και ενα για να το τερματισει.

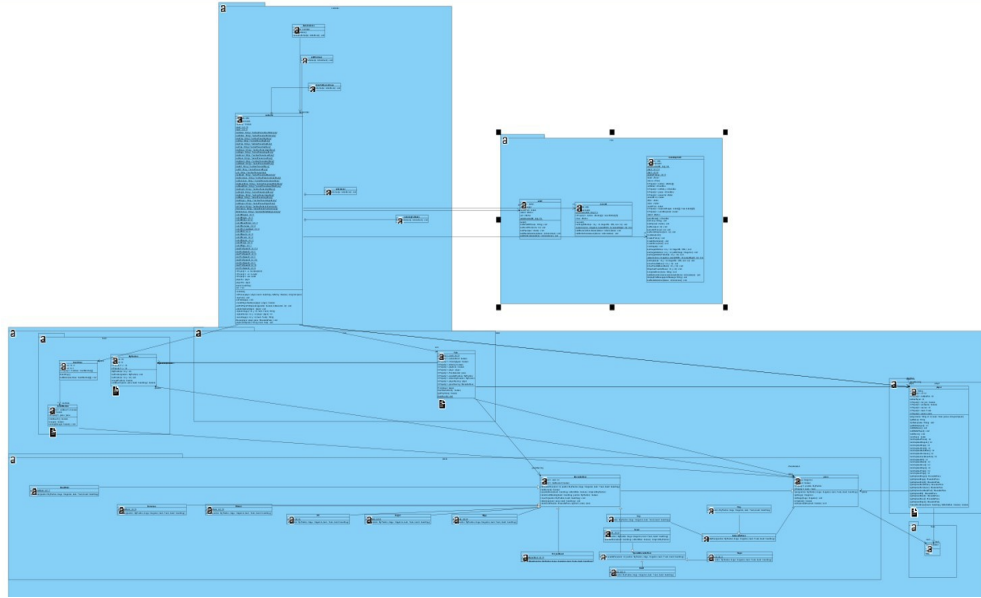
Και αυτο οι μονες συναρτησεις που εχει ειναι setters και συναρτησεις για να προσθεσουν actionListeners

παρακατω φαινεται το διαγραμμα του view:



5. Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Παρακατω φαιναιτο το ολοκο διαγράμματα:



Εδω φαίνεται ακομα πιο ξεκαθαρα οτι το view με το model δεν εχουν καμια ενωση και ειναι μονο ο controller που εννονεται και στα δυο και τα χρησιμοποιει. Οι περισσοτερες εννωσεις ειναι στο model οπου βλεπουμε οτι πολλες απο τις κλασεις του να χρησιμοποιουνται σε πολλες αλλες ειντε σαν πεδια ειντε σαν παραμετρους συναρτησεις.

6. Λειτουργικότητα (Β Φάση)

Πιστευω πως καταφερα να υλοποιησω ολα τα ζητουμενα ερωτημα, σθν καπες επιπλεων λεπτομεριες που θεωρησα σκοπημες.

7. Συμπεράσματα

Ηταν πολυ ειδηαιτερο αυτο το project καθως ηταν η πρωτη φορα που υλοποιησα μονος μου ενα project αυτου του μεγεθους, η πρωτη φορα που χρησιμοποιησα γραφικα στην java αλλα και γενικα. Τελος ηταν η πρωτη φορα που ασχοληθηκα με το mvc μοντελο. Το τελευταια το βρηκα ιδιατερα βοηθητικο και χρησιμο καθως ολες οι ιδιοτητες ηταν ξεχωριστε μεταξ η του και ηταν ιδιατερα εθκολο να προσθετεις κενουργια πραγματα και να βελτιωνεις ηδη υπαρχοντα. Στην αρχη ηταν λιγο δυσκολο μεχρι να καταλαβω την λογικη και του που θα επρεπε να βαζω τι καθως πριν απο σε ολες τις εργασιες ολα ηταν τις περισσοτερες φορες σε ενα αρχαιο και δεν υπηρχε αυτος ο διαχωρισμος. Πιστευω οτι η λογικη και αυτα που εμαθα απο το mvc θα μου μηνουν και θα τα χρησιμοποιω καθε φορα που ασχολουμε με γραφικα.

Επισεις μου εκανε αρκετη εντηπωση το ποσες αλλαγες εκανα απο την πρωτη φαση. Το περιμενα οτι θα κανω αλλαγες αλλα εβρησκα τον εαυτο που πολλες φορες να αλαζει

ολοκληρη την λογικη που ειχα σκευτει καθώς εβλεπα πως θα βολευει οσο εξελιζοταν το προγραμμα.

Γενικα μπορω να πω πως ηταν με διαφορα το ποιο ενδιαφερον project που εχω κανει και ενα αρκετα ωραιο 'διαλημμα ' απο την ομιλιπες ασκησεις καθώς απολαμβανα ιδιαιτερα να δουλεω σε αυτο.