



Πανεπιστήμιο Κρήτης-Τμήμα Επιστήμης Υπολογιστών
HY252-Αντικειμενοστραφής Προγραμματισμός

Διδάσκων: Ι.Τζιτζίκας
Χειμερινό Εξάμηνο 2022-2023

Stratego Ice vs Fire



ΚΑΣΤΡΙΝΑΚΗΣ ΙΩΑΝΝΗΣ
CSD 4647
13/01/2023

Περιεχόμενα:

1.	Εισαγωγή.....	1
2.	Η Σχεδίαση και οι Κλάσεις του Πλακέτου Model.....	1
3.	Η Σχεδίαση και οι Κλάσεις του Πλακέτου Controller.....	1
4.	Η Σχεδίαση και οι Κλάσεις του Πλακέτου View.....	2
5.	Η Αλληλεπίδραση μεταξύ των κλάσεων - Διαγράμματα UML.....	2
6.	Λειτουργικότητα(Β Φάση).....	2
7.	Συμπεράσματα.....	2

1. Εισαγωγή

Στην υλοποίηση αυτής της εργασίας θα χρησιμοποιήσουμε το προγραμματιστικό μοντέλο MVC (model view controller). Συγκεκριμένα, θα χωρίσουμε το πρόγραμμα μας σε τρία μέρη που το καθένα θα παίζει έναν διαφορετικό ρόλο.

Αρχικά, το Model είναι αυτό, το οποίο θα περιλαμβάνει όλη την επεξεργασία δεδομένων, θα κάνει τους απαραίτητους ελέγχους και ουσιαστικά μέσα σε αυτό θα υπάρχουν όλα τα τεχνικά μέρη που χρειάζονται ώστε να λειτουργήσει το παιχνίδι με τον σωστό τρόπο.

Έπειτα, το View είναι αυτό, το οποίο θα είναι υπεύθυνο για την οπτική απεικόνιση του προγράμματος στον χρήστη, δηλαδή, το γραφικό περιβάλλον του παιχνιδιού.

Τέλος, έρχεται ο Controller, αναπαριστώντας τον εγκέφαλο του προγράμματος και εκτελώντας τις κατάλληλες ενέργειες ώστε να συνδέσει το Model με το View.

Σημειώση από φαση 2 : Γράφοντας την αναφορά και τα javadocs συνειδητοποίησα οτι η υλοποιηση μου δεν ειναι ακριβως οπως θα επρεπε να ειναι ενος μοντελου mvc . Θα επρεπε η λογικη περα σε οτι αναφερεται για JPanel JFrame κτλ να βρίσκεται στον controller (π.χ. το movement του piece) και να καλω με συναρτησεις να γινονται οι αλλαγες στο γραφικο περιβαλλων. Άλλα λογω φορτου δεν μπορω να το διορθωσω τωρα

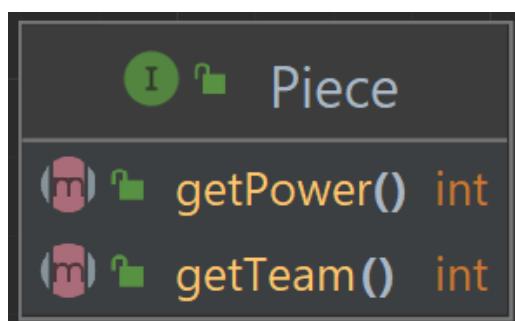
2. Η Σχεδιαση και οι Κλασεις του Πακετου Model

Σε αυτό το πακέτο θα περιλαμβάνονται τα πακέτα με τις κλάσεις board, collections, fight, την διεπαφή piece και την abstract κλαση player.

PACKAGE Piece

Μεσα στο πακετο piece εχουμε μια διεπαφη piece και 2 πακετα MovablePieces και ImmovablePieces.

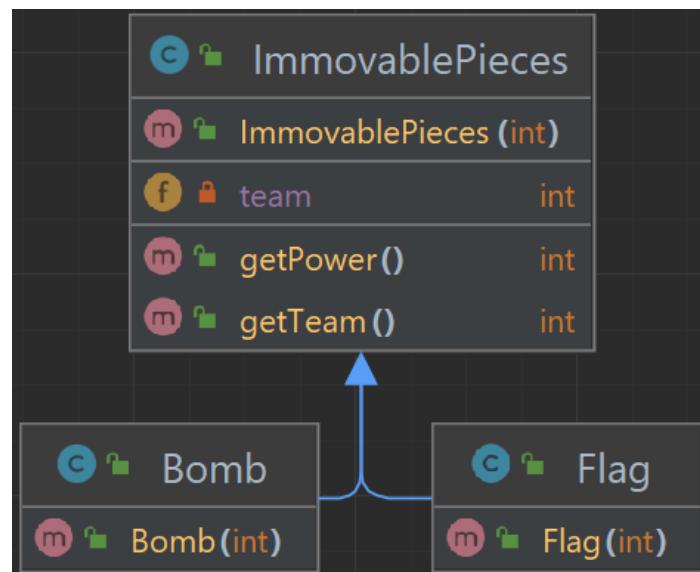
Interface Piece



Ξεκινωντας, στην διεπαφή Piece θα περιλαμβανονται ολες οι μεθόδοι που θα χρησιμοποιησουμε για όλα τα πιονια ανεξαρτητως αν κινουνται η οχι .

1. int getPower() επιστρεφει την δυναμη του πιονιου
2. int getTeam() επιστρεφει την ομαδα του πιονιου

```
PACKAGE ImmovablePieces/
class ImmovablePieces
```



Η κλαση ImmovablePieces ειναι μια υποκλάση της διεπαφης Piece. αποτελείται απο :

την ιδιωτικη μεταβλητη -> int team : οπου ειναι η ομαδα που ειναι το πιονι

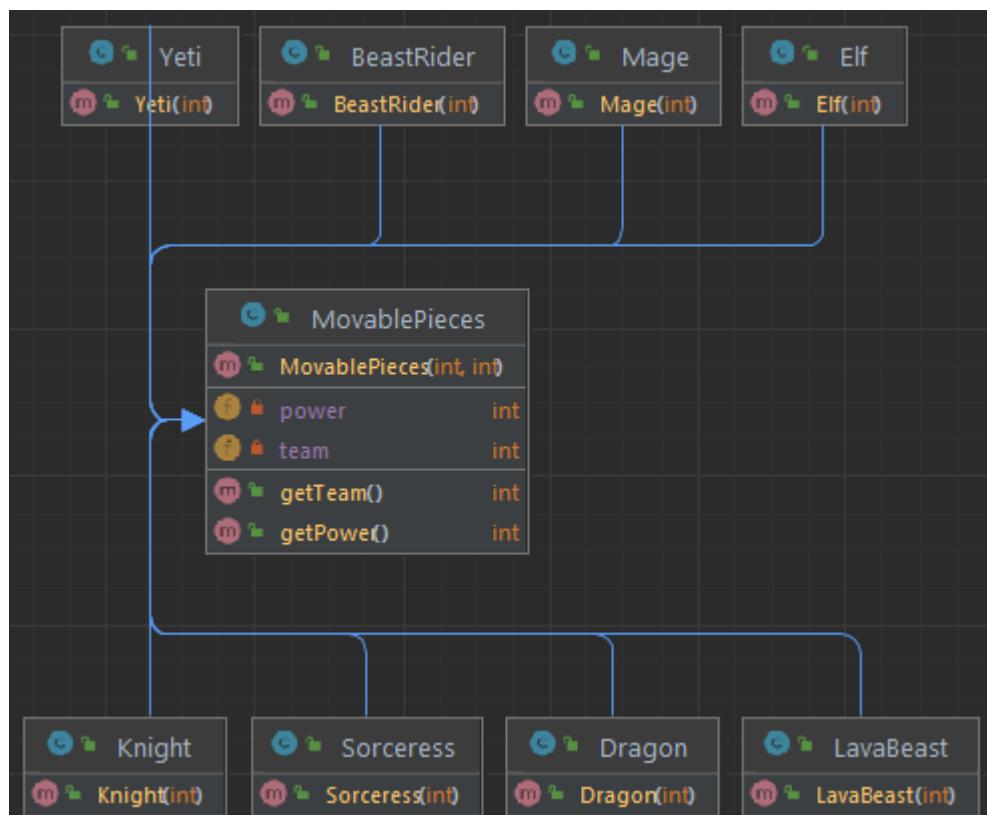
εναν constructor -> public ImmovablePieces : δημιουργει το profile του πιονιου (δυναμη=0 εχει και τι ομαδα ειναι)

και τις μεθοδους -> public int getPower() : επιστρεφει 0 (αυτα τα πιονια δεν εχουν δυναμη)
public int getTeam() : επιστρεφει την ομαδα του πιονιου

Επισης η ImmovablePieces εχει 2 υποκλασεις την Bomb και την Flag οι οποιες και οι 2 εχουν απο εναν constructor ο οποιος καλει με το key super τον constructor την κλασης ImmovablePieces και μεσω αυτου αποθηκεύουν τις τιμες τους.



PACKAGE MovablePieces/
class MovablePieces



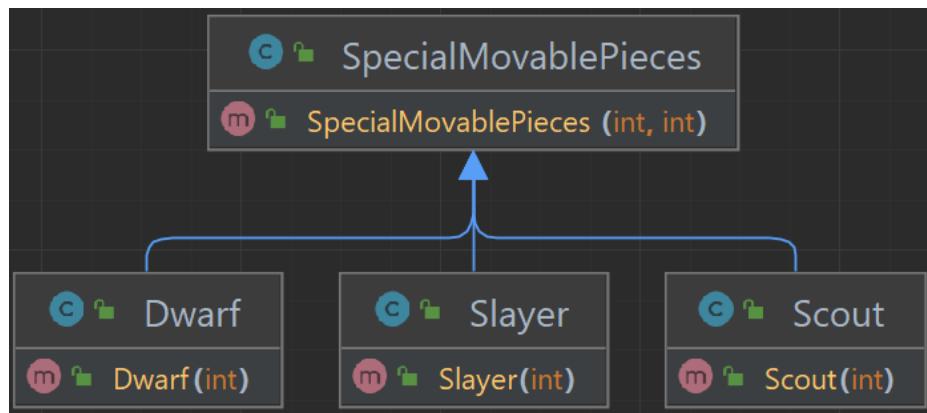
Η Κλαση MovablePieces ειναι και αυτη επισης μια υποκλάση της Piece, και αποτελειται απο:

1. τις ιδιωτικες μεταβλητες -> int power : η δυναμη του piece
int team : η ομαδα του piece
2. εναν constructor -> public MovablePieces : δημιουργει το profile του πιονιου (ποση δυναμη έχει και τι ομαδα ειναι)
3. τις μεθοδους -> public int getPower() : επιστρεφει την δυναμη του πιονιου
public int getTeam() : επιστρεφει την ομαδα του πιονιου

οι οποίες όπως παραπάνω, υλοποιούν τις υπογραφές των συναρτησεων που βρισκονται στο interface Piece.

Οπως φαίνεται και από την εικόνα περιεχει 8 υποκλασεις που η καθε μια αναπαριστα ενα πιονι. Και οι 8 εχουν απο εναν constructor ο οποιος καλει με το key super τον constructor την κλασης MovablePieces και μεσω αυτου αποθηκεύουν τις τιμες τους.

Επιπλεων αυτο το πακετο περιεχει και ενα πακετο specialMovablePieces που μεσα εχει μια υποκλαση της movablePiece :

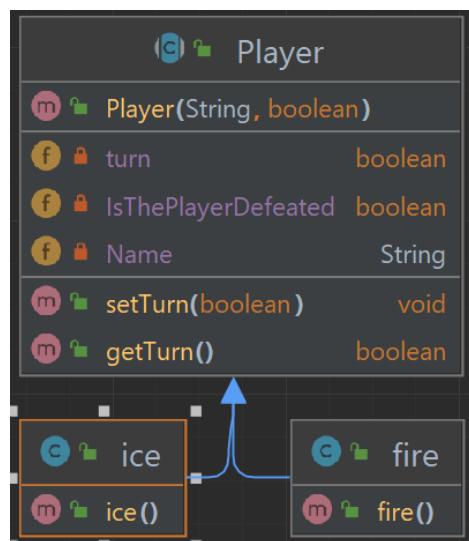


Αποτελειται απο ενα constructor -> SpecialMovablePieces(power,team) : που με το key super δινει τις τιμες power και team στον constructor της movablePieces

Εχει 3 υποκλασεις : dwarf,slayer και scout που αναπαριστούν 3 special movable pieces. Και αυτες εχουν constructor που δουλευει με τον ίδιο τροπο με αυτους των απο πανω

PACKAGE Player

Αυτο το πακετο περιεχει την abstract κλαση player ,και τις δυο υποκλασεις της, Ice και Fire, που αναπαριστούν τις 2 ομαδες

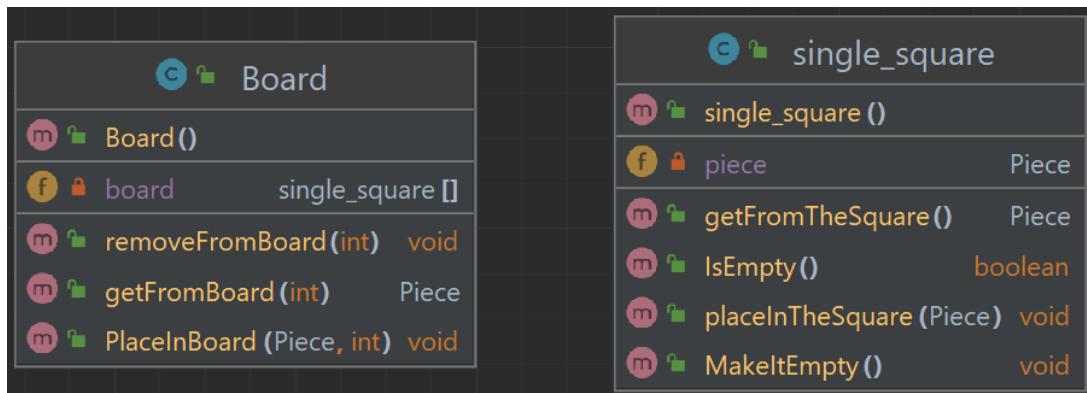


Η Abstract κλαση Player ειναι αυτη που ουσιαστικα αναπαριστα τον παιχτη. Αποτελειται απο :

1. Εναν constructor public Player(String name,boolean turn) ο οποιος φτιαχνει το profile του παικτη με βαση το ονομα και την σειρα.
2. Τις ιδιωτικες μεταβλητες turn (αν ειναι σειρα του παικτη η οχι), IsThePlayerDefeated (αν ο παιχτης εχει χασει η οχι) και Name (ονομα ομαδας)
3. Και τις 2 μεθοδους getName() (επιστρεφει το ονομα της ομαδας) και getTurn() (επιστρεφει αν εχει σειρα η οχι)

Οπως φαίνεται και στο διαγραμμα απο πανω, η Player εχει και 2 υποκλασεις, την **ice** και την **fire** που η καθε μια εχει εναν constructor, και ετσι καθε μια αναλογως δημιουργει ειτε την ομαδα ice ειτε την ομαδα fire

Package Board/ Class Board & Class single_square



Ξεκιναμε με την κλαση Board.Η κλαση Board ειναι αυτη που θα αναπαραστήσει το ταμπλο του παιχνιδιου, που πανω του θα μετακινούνται τα πιόνια.

Αποτελειται απο :

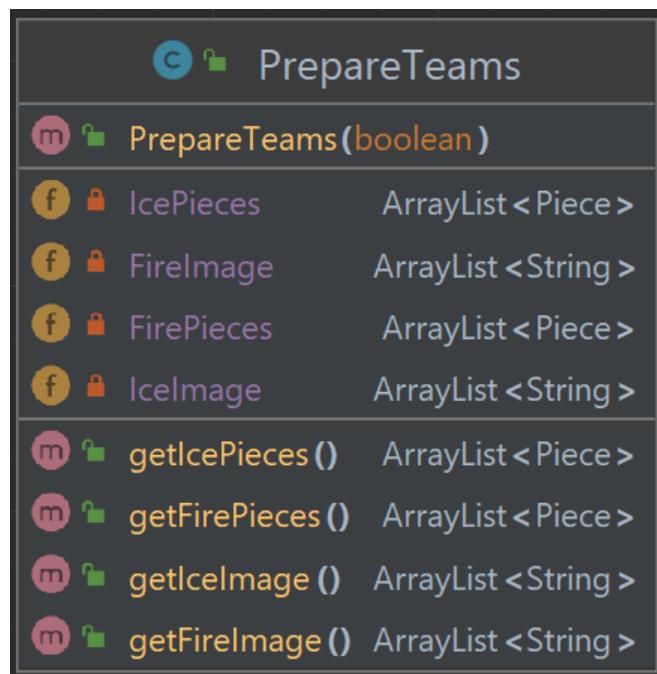
1. Εναν ιδιωτικο πίνακα 80 θεσεων τυπου `single_square` (θα μιλησουμε σε λιγο για αυτο) ο οποιος αναπαριστα το ταμπλο του παιχνιδιου
2. Εναν constructor public `Board()` (ο οποιος αρχικοποιει αυτον τον πινακα(τις θεσεις του))
3. 3 συναρτησεις,->Μια συναρτηση public `piece` `getFromBoard(int coordinates)` η οποια με βαση τις συντεταγμενες που τις δινονται επιστρεφει το πιονι που βρίσκεται σε εκεινο το σημειο (αν δεν βρίσκεται επιστρεφει null), -> Μια συναρτηση public void `PlaceInBoard(Piece piece, int coordinates)` η οποια τοποθετει το πιονι που της δινεται στις συντεταγμενες που επισης της δινονται και -> Μια συναρτηση public void `removeFromBoard(int coordinates)` η οποια με βαση τις συντεταγμενες που τις δινονται βγαζει ενα πιονι απο εκεινο το σημειο(το θετει null)

Από την αλλη , η κλαση single_square αναπαριστα ενα τετραγωνο απο το ταμπλο.

Αποτελειται απο :

1. Την ιδιωτικη μεταβλητη piece τυπου Piece που αναπαριστά ενα πιονι
2. Εναν Constructor single_square() που θετει το τετραγωνο κενο
3. 4 συναρτησεις,-> MakeIsEmpty() που και αυτη θετει το τετραγωνο κενο
-> getFromTheSquare() που επιστρεφει το πιονι που ειναι πανω στο τετραγωνο(αν ειναι αδειο επιστρεφει null,->IsEmpty() που επιστρεφει αν το κενο ειναι αδειο η οχι,
->placeInTheSquare(Piece piece) που τοποθετει στο τετράγωνο το piece που της δινεται

Package Collections/ Class PrepareTeams

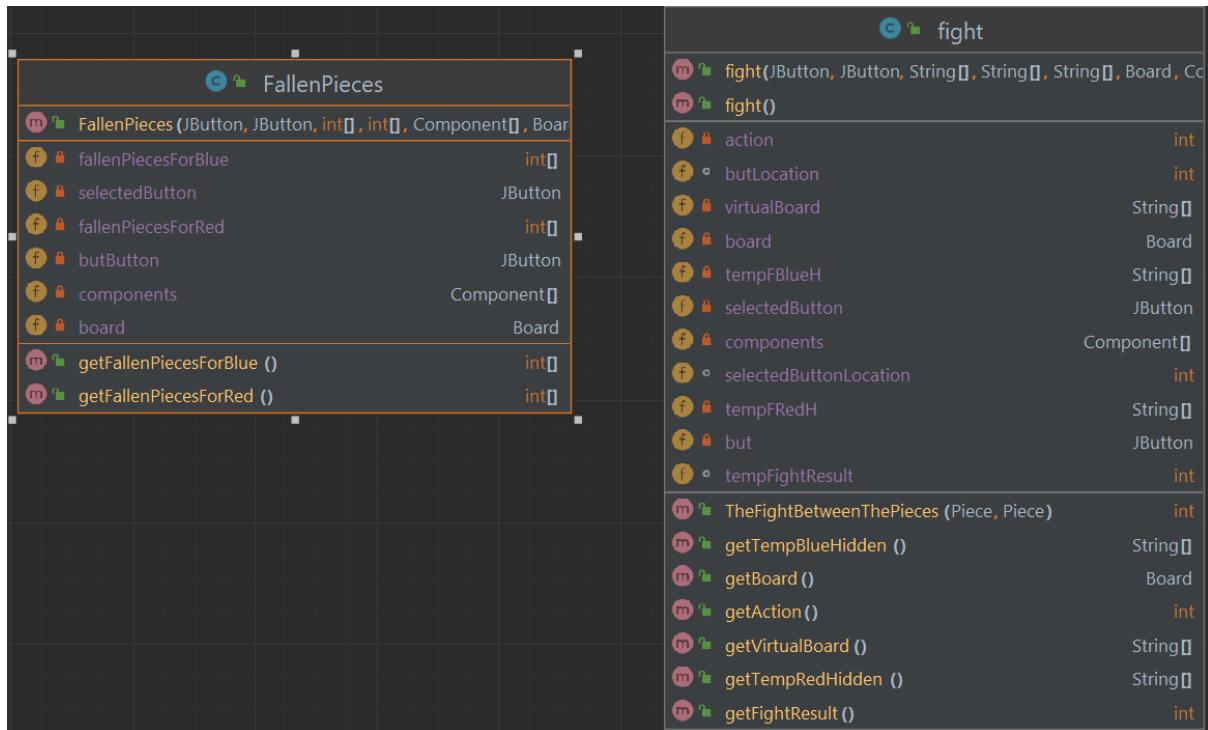


Σε αυτην την κλαση δημιουργουνται ουσιαστικα οι 2 ομαδες.Η κλαση αποτελειται απο:

1. Τις 4 ιδιωτικες ArrayLists -> 2 τυπου piece (για τα πιονια),μια για καθε ομαδα (Ice,Fire) και -> 2 τυπου string (για τις εικονες των πιονιων) ,μια για καθε ομαδα (Ice,Fire)
2. Τον constructor `PrepareTeams()` που δημιουργει σε καθε ομαδα τα πιονια της
3. Και τις 4 συναρτησεις `getIcePieces()`, `getIcelImage()`, `getFireImage()` και `getFirePieces()` για να επιστρεφει τα πιονια της καθε ομαδας και τα ονομα των εικονων τους

Package Fight

Αυτό το πακέτο περιέχει 2 κλασεις που αφορούν την μαχη μεταξυ 2 πιονιων. Την fallenPieces και την fight



FallenPieces : Αυτη η κλαση χρησιμοποιείται για την καταμετρηση των πιονιων τα οποια εχουν χασει ωστε να απεικονιστουν στο side panel

Αποτελείται απο :

1. εναν constructor : ο οποιος περνει σαν ορισματα -> το κουμπι που πατησε ο χρηστης ωστε να επιλεξει το πιονι που θα κουνηθει ,το κουμπι που δειχνει στο σημειο που θα παει αυτο το πιονι,τους 2 πινακες int που μετρανε τα νικημενα πιονια της καθε ομαδας, τα components του panel που ειναι πανω το ταμπλο του παιχνιδιου ωστε με την βοηθεια και των κουμπιων να βρει για ποια πιονια απευθύνεται ,και τελος τον πινακα που ειναι πανω τα πιονια
2. τις ιδιωτικες μεταβλητες -> fallenPiecesForBlue, fallenPiecesForRed, selectedButton, butButton, components και board , που χρησιμοποιουνται για να θέσουμε σε αυτα τα ορίσματα του constructor
3. 2 συναρτησεις -> getFallenPiecesForBlue() και getFallenPiecesForRed() για να επιστρεφουν τους ανανεωμενους πινακες των νικημενων πιονιων της καθε ομαδας

fight : Σε αυτην την κλαση ουσιαστικα γινεται η κινηση του πιονιου αν μετακινηθει σε κενο τετραγωνο η αλλιως η μαχη μεταξυ των 2 πιονιων (αυτου που επιλεχθηκε και αυτου που βρησκετε στο τετραγωνο που επελεξε ο χρηστης να παει το πιονι)

Αποτελείται απο :

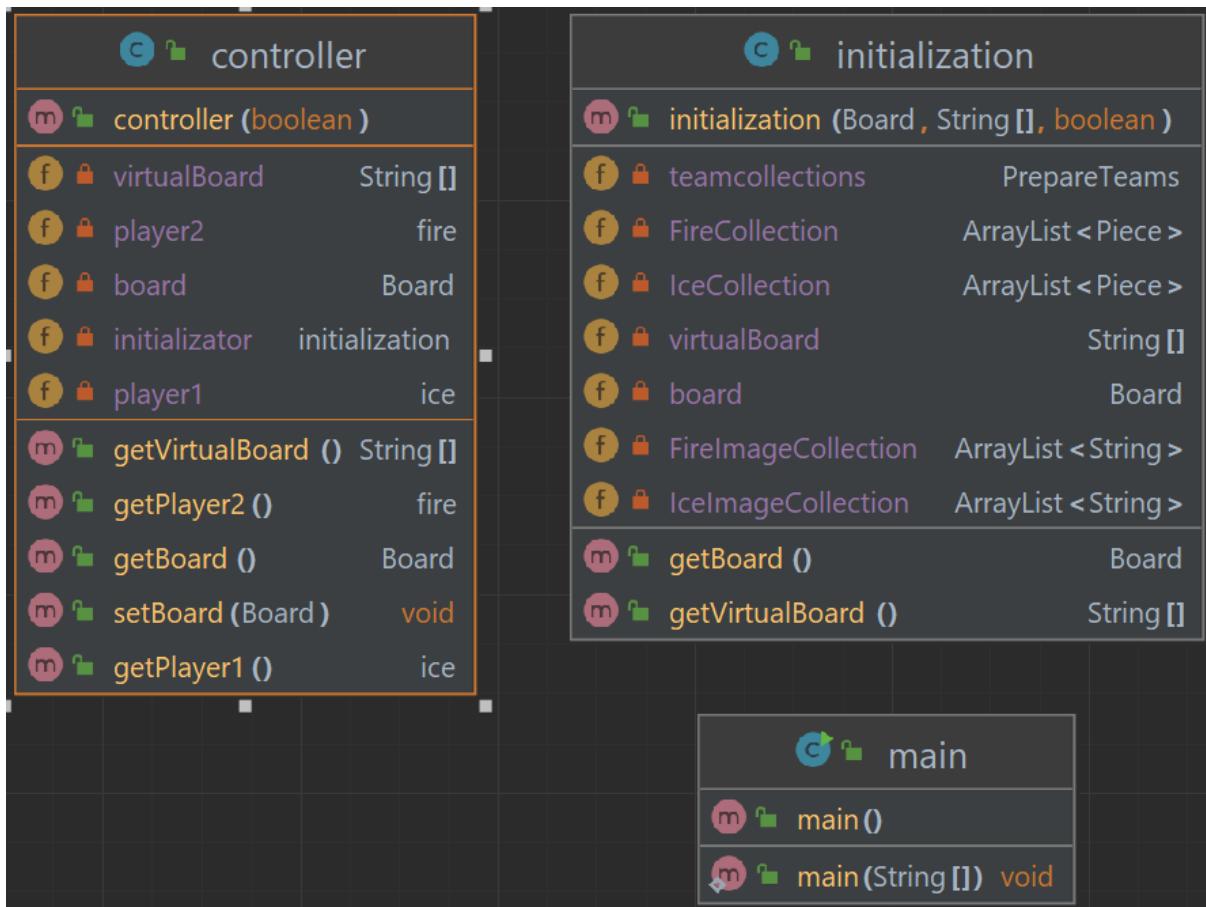
1. 2 constructor -> εναν χωρις ορισματα (fight()), για όταν θελω δω τα αποτεσματα της μαχης (συναρτηση TheFightBetweenThePieces) χωρις να αποθηκευσω τιποτα, και εναν με ορισματα->το κουμπι που πατησε ο χρηστης ωστε να επιλεξει το πιονι που θα κουνηθει ,το κουμπι που δειχνει στο σημειο που θα παει αυτο το πιονι,το virtual board που περιεχει τα ονοματα των εικονων των πιονιων,το tempFBlueH που περιεχει τα ονοματα των εικονων των μπλε πιονιων,το tempFRedH που περιεχει τα ονοματα των εικονων των κοκκινων πιονιων, τα components του panel που ειναι πανω το ταμπλο του παιχνιδιου ωστε με την βοηθεια και των κουμπιων να βρει για ποια πιονια απευθυνεται ,και τελος τον πινακα που ειναι πανω τα πιονια
2. τις ιδιωτικες μεταβλητες που φαίνονται και στην εικονα, και χρησιμοποιουνται για να θέσουμε σε αυτες τα ορισματα του constructor
3. 7 συναρτησεις -> int TheFightBetweenThePieces (Piece A , Piece B) οπου υπολογιζει το αποτελεσμα της μαχης μεταξυ των δυο πιονιων που περνει σαν ορισματα, int getAction() οπου επιστρεφει αν τα pieces ειναι ιδιας ομαδας η οχι, Board getBoard() οπου επιστρεφει τον ανανεωμένο πινακα, String[] getVirtualBoard() οπου επιστρεφει τον virtual board ανανεωμένο , String[] getTempBlueHidden() οπου επιστρεφει τον πινακα για τα ονοματα των εικονων των μπλε πιονιων, String[] getTempRedHidden() οπου επιστρεφει τον πινακα για τα ονοματα των εικονων των κοκκινων πιονιων ,και την getFightResult() οπου επιστρεφει το αποτελεσμα της μαχης

3. Η Σχεδιαση και οι Κλασεις του Πακετου Controller

Ο εγκεφαλος του παιχνιδιου

Αποτελειται απο τις κλάσεις controller, initialization και main

Package controller/
 Class controller , Class initialization
 & Class main



main : χρησιμοποιείται μόνο για την εναρξη του παιχνιδιου

initialization : αυτη η κλαση χρησιμοποιείται για την τυχαια τοποθετηση των πιονιων στο ταμπλο ,και των ονοματων των εικονων των πιονιων στον virtual board.

Αποτελείται απο :

1. εναν constructor με ορίσματα -> τον board ,τον virtual board και την boolean μεταβλητη REDUSEDARMY για το αν θα δημιουργησει 15 πιονια για καθε ομαδα ή 30
2. τις ιδιωτικες μεταβλητες που φαίνονται στη εικονα απο πανω -> 4 λιστες για να αποθηκευσω αυτες που πήρα απο την PrepareTeams , 1 τυπου PrepareTeams για να δημιουργηθουν οι αμαδες, 1 τυπου board για να μπουν τα pieces ,και 1 τυπου virtual board να μπουν τα ονοματα των εικονων των pieces
3. 2 συναρτησεις -> getBoard() και getVirtualBoard() που επιστρεφουν τους γεματους πλεων board kai virtualBoard

controller : αυτη η κλαση δημιουργει το παιχνιδι.

Αποτελείται απο :

1. εναν constructor controller(boolean REDUSEDARMY) ο οποιος δημιουργει τις 2 ομαδες , το board με τα pieces, τον virtual board με τα ονοματα των εικονων των pieces,βαση την τιμη

του ορισματος REDUSEDARMY κανει initialize τα 2 boards ειτε με 30 pieces για την καθε ομαδα ειτε με 15, και θετει τους καινουργιους πινακες για να ξεκινησει το παιχνιδι.

2. τις ιδιωτικες μεταβλητες που φαινονται στην εικονα απο πανω ωστε να μπορει ο constructor να δουλεψει
3. 5 συναρτησεις ->getBoard() επιστρεφει τον πινακα με τα pieces, getVirtualboard() επιστρεφει τον πινακα με τα ονοματα των εικονων των pieces, getPlayer1() επιστρεφει τον παικτη 1 ,getPlayer2() επιστρέφει τον παικτη 2, και την setBoard(board) που θετει στον board του controller τον board που της εχει δοθει σαν ορισμα

4. Η Σχεδιαση και οι Κλασεις του Πακετου View

Εδω ουσιαστικα γινεται οτι αφορα τα γραφικα του παιχνιδιου

PACKAGE View/ Class Menu , Class normalGame & sidePanel

Το πακετο της View περιλαμβανει τις εικονες που θα χρησιμοποιησουμε , την κλαση sidePanel η οποια θα δημιουργει το side panel που θα κανει attach στο main frame, την κλαση Menu που δημιουργει το main menu το πρωτο πραγμα που θα βλεπει ο χρηστης οταν ανοιγει το προγραμμα και απο εκει θα διαλεγει ποιο gamemode θελει πριν πατησει να ξεκινησει το κανονικο παιχνιδι ,και τελος την κλαση normalGame στην οποια δημιουργειται το main frame και διαχειριζεται την καθε επιλογη του παιχτη απο εκει και περα.

Menu

Menu		
m Menu()		
	isTheRedusedGameTrue	boolean
	startNoStepBack	JButton
	startReduced	JButton
	isTheNoStepBackTrue	boolean
	checkBox2	JButton
	startNormal	JButton
	backgroundImage	JLabel
	checkBox1	JButton
	StartReducedGameButton ()	void
	actionPerformed (ActionEvent)	void
	BackgroundImage ()	void
	menuFrame ()	void
	StartNormalGameButton ()	void
	StartNoStepBackGameButton ()	void
	BuildTheCheckBox1 ()	void
	BuildTheCheckBox2 ()	void

Menu : δημιουργει το main menu, το πρωτο πραγμα που θα βλεπει ο χρηστης οταν ανοιχει το προγραμμα και απο εκει θα διαλεγει ποιο gamemode θελει πριν πατησει να ξεκινησει το κανονικο παιχνιδι.



Αποτελειται απο :

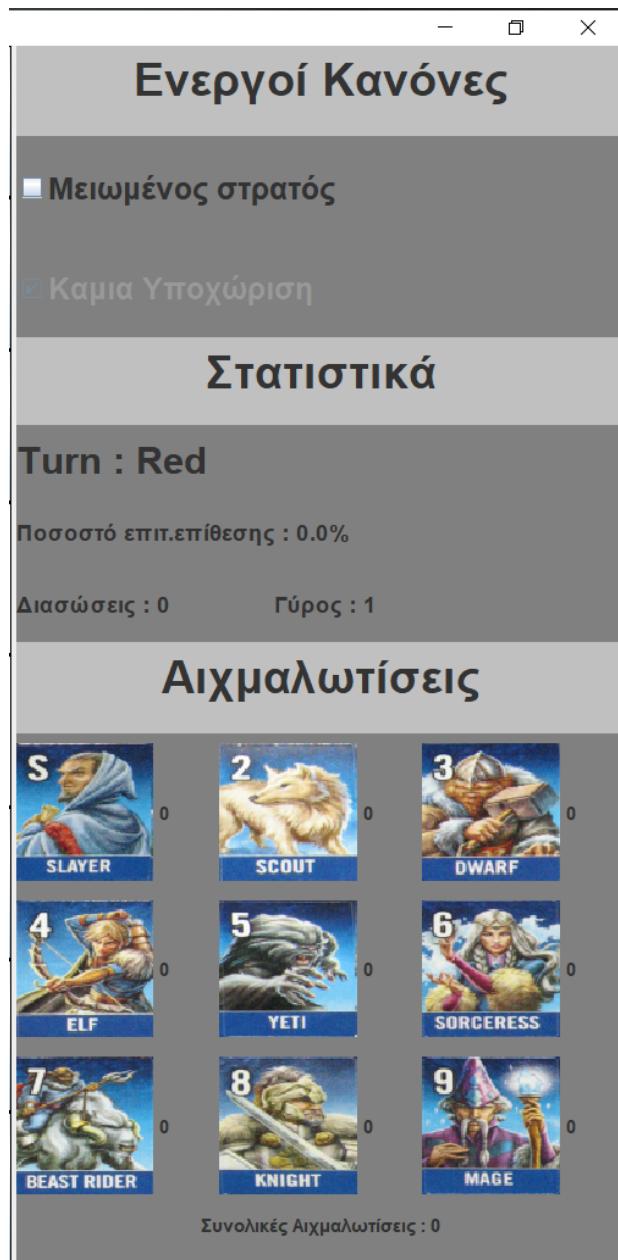
1. evan constructor που δημιουργει με την βοηθεια των αλλων συναρτησεων το Μενου
2. 8 ιδιωτικες μεταβλητες -> 5 τυπου jbutton που θα ειναι τα κουμπια απο τα οποια θα επιλέγει ο χρηστης τι θελει, μια τυπου jlabel οπου εκει θα μπαινει η εικονα και τα κουμπια ,και 2 boolean ,μια isTheReducedGameTrue που υποδηλωνει αν ο χρηστης επελεγε να παιξει με μισο στρατο η καθε ομαδα ,και μια isTheNoStepBackTrue που υποδηλωνει αν ο χρηστης επελεξε να παιξει χωρις να μπορουν τα πιονια να κανουν βημα προς τα πισω
3. και 8 συναρτησεις -> 5 για την δημιουργια των κουμπιων,1 για την δημιουργια της background εικονας, 1 για την δημιουργια του jframe που θα μπουν τα παραπανω ,και 1 actionPerformed η οποια μολις πατησει ο χρηστης ενα κουμπι το λαμβανει και πραττει αναλογως

sidePanel

sidepanel

m	sidepanel (JFrame, int[], int[], boolean, boolean)	
f	sidePanel1	JPanel
f	sidePanel5	JPanel
f	sidePanel6	JPanel
f	sidePanel3	JPanel
f	fallenPiecesForBlue	int[]
f	fallenPiecesForRed	int[]
m	getFallenPiecesForBlue ()	int[]
m	getFallenPiecesForRed ()	int[]

sidePanel : δημιουργεί το side panel το οποιο τοποθετείται στο main Jframe μολις το παιχνιδι ξεκινησει (μετα το Μενου).Δειχνει διαφορες πληροφοριες στην σειρα του κάθε παίκτη για αυτον, καθως και γενικες πληροφοριες για το παιχνιδι (ποιο gamemode διαλεξε ,ποιο round ειναι etc)



Αποτελείται από :

1. εναν constructor που δημιουργει το jpanel που θα μπει στο main jframe ,και επισης δημιουργει 7 JPanels που κανουν attach πανω το main JPanel.Το καθενα απο αυτα αναφερεται σε κατι διαφορετικο για τον χρηστη.Εκει που ειναι οι εικονες δεν εχω βαλει τον δρακο για λογους ομοιομορφιας αλλα τον υπολογιζω κανονικα
2. 6 μεταβλητες -> 4 public για να μπορω να ενημερωνω τα στοιχεια που θελω κατα την διαρκεια του παιχνιδιου ,και 2 για τα fallen pieces καθε ομαδας
3. 2 συναρτησεις (getters) για να επιστρεφουν τους πινακες με τα fallen pieces καθε ομαδας

NormalGame

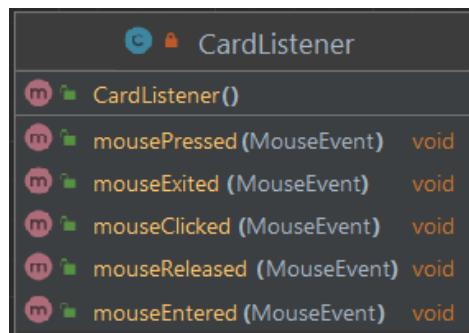
NormalGame		
m	NormalGame (boolean , boolean)	
f	ActualGame controller	
f	fallenPiecesForRed int[]	
f	NOSTEPBACK boolean	
f	cardname String	
f	tempForHiddenBLUE String []	
f	panel JPanel	
f	selectedButton JButton	
f	REDUSEDARMY boolean	
f	fallenPiecesForBlue int[]	
f	virtualBoard String []	
f	tempForHiddenRED String []	
f	Round int	
f	cl CardListener	
f	blueSuccessAttack int	
f	components Component []	
f	TheSidePanel sidepanel	
f	redAttack int	
f	blueAttack int	
f	redSuccessAttack int	
f	iconSelected boolean	
m	caseForScout (int, int) boolean	

NormalGame : δημιουργει το jframe και το jpanel οπου θα αναπαραστατε το ταμπλο των παιχνιδιου και διαχειριζεται την καθε επιλογη του χρηστη απο εκει και περα.



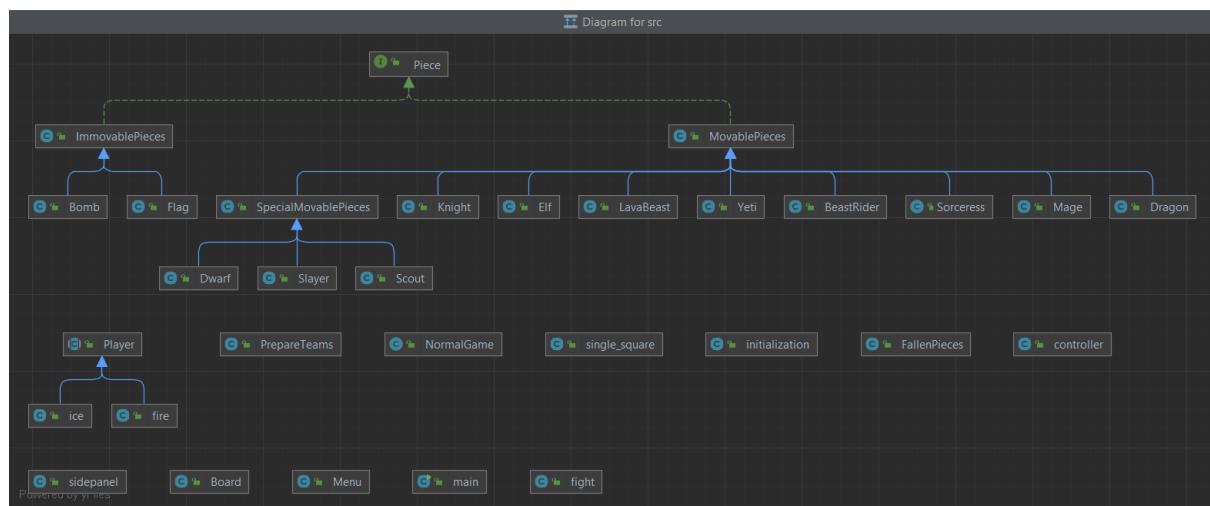
Αποτελείται από :

1. τον constructor NormalGame (boolean,boolean) που δημιουργεί αναλογά το παιχνίδι για την gameMode επελεξε ο χρηστης ,το JFrame,το JPanel οπου θα μπει το ταμπλο και ουσιαστικα ξεκιναει το παιχνιδι
2. την συναρτηση caseForScout(int,int) που επιστρεφει boolean τιμη. Αν το πιονι που επιλεχθηκε να μετακινηθει ειναι ο scout επιστρεφει αν μπορει να παει εκει που διαλεξε ο χρηστης η οχι.
3. τις ιδιωτικες μεταβλητες που φαινονται παραπανω
4. την κλαση CardListener οπου κανει implement την MouseListener



Εκει ουσιαστικα γινεται οτι χρειαζεται μολις ξεκινησει το παιχνιδι αναλογα με το τι επιλεγει ο χρηστης καθε φορα μεχρι το τελος του παιχνιδιου (ολη η λογικη που χρειαζεται για να γινει κινηση).

5. Η Αλληλεπιδραση μεταξυ των κλασεων - Διαγραμματα UML



6. Λειτουργικοτητα (Β Φαση)

Το παιχνίδι λειτουργει κανονικα (περα από τον μηχανισμο του resurrection). Μεσω του μενου μπορει ο χρηστης να επιλεξει αν θελει να παιξει με μειωμενο στρατο ή χωρις να μπορει μετακινηθει προς τα πισω ή και τα 2. Το side panel λειτουργει επισης κανονικα με μονη εξαιρεση οτι δεν εμφανιζει ποσα πιονια εχουν διασωθει γιατι οπως προανεφερα δεν εφτιαξα τον μηχανισμο του resurrection.Οταν ο χρηστης παταει ενα πιονι ΔΕΝ εμφανιζει που μπορει να παει αλλα τα ορια για την κινηση τον πιονιων εχουν τεθει και ειναι σωστα ακομα και για τον Scout (o scout μπορει να παει οσα τετραγωνακια θελει μπροστα ,πισω ,δεξια και αριστερα αρκει στην διαδρομη μεχρι τον προορισμο να μην υπαρχει κανενα πιονι.Σχετικα με τις κιτρινες περιοχες τον αφησα να περναει καθως δεν βρηκα καπου να λεσι οτι δεν μπορει και για να ειναι ποιο ενδιαφερων και το παιχνιδι, καθως να προσθεσω να μην πηγαινει και στα κιτρινα ειναι αρκετα ευκολο απο την στιγμη που το υπολοιπο παιχνιδι δουλευει σωστα).

Τωρα σε σχεση με ενα προβλημα που ανακαλυψα γραφοντας την αναφορα, το MVC μοντελο μου δεν ειναι αρκετα σωστο.Έχω κατανοησει πως θα επρεπε να ειναι αλλα καθως εγραφα τον κωδικα λογο φορτου απλα προσθετα καθε μερα και κατι στον κωδικα ξεχωντας η λογικη να διαδρματιζεται στον controller.Αυτο που θα αλλαζα αν μπορουσα να διαθεσω παραπανω χρονο θα ηταν ο κωδικας που εχει σχεση με την λογικη που βρισκεται μεσα στην cardListener να ηταν στον controller και απο εκει να καλειται οτι χρειαζεται σε μορφη συναρτησης απο την view σχετικα με την εικονα και το ti βλεπει ο χρηστης.Πιστευω όλο το model ειναι σωστο αν και καποια οπως η δυναμη του πιονιου δεν χρησιμοποιηθηκε αφου το εκανα με “instanceof” αλλα το αφηνω καθως μπορει να γινει και αλλιως περα απο τον τροπο που εκανα.

7. Συμπερασματα

Η εργασια για να γινει σωστα χρειαζεται αρκετο χρονο αλλα πιστευω οτι ηταν πολυ χρησιμη ωστε να μαθω να δουλευω σε ενα project (εμαθα τι να κανω και τι να μην κανω),και πηρα αρκετες γνωσεις για το πως δουλευουν πολλα πραγματα στην java και οχι μονο.