

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστραφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2020-2021**



*Εισαγωγή*

Think and describe what you plan to do and why it will be useful.

**Περιεχόμενα**

[**1.**](#_heading=h.gjdgxs) **Εισαγωγή** 2

[**2.**](#_heading=h.30j0zll) **Η Σχεδίαση και οι Κλάσεις του Πακέτου Model** 2

[**3.**](#_heading=h.1fob9te) **Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller** 5

[**4.**](#_heading=h.3znysh7) **Η Σχεδίαση και οι Κλάσεις του Πακέτου View** 6

[**5.**](#_heading=h.2et92p0) **Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML** 6

## 

## **Εισαγωγή**

Για την υλοποίηση της εργασίας ακολουθείται το μοντέλο MVC (Model View Controller). Ουσιαστικά το Model περιέχει όλα τα στοιχεία του παιχνιδιού , το View τα απεικονίζει και ο Controller ενώνει το Model και το View. Παρακάτω αναλύονται κυρίως το Model και ο Controller και λιγο το View.

## **Η Σχεδίαση και οι Κλάσεις του Πακέτου Model**

Σε αυτό το πακέτο περιέχονται η διεπαφή Piece , οι κλάσεις MovablePiece , ImovablePiece, SpecialMovablePiece, Player και το Board.

**Piece Interface and classes that implements that interface**

Η διεπαφή Piece περιέχει ότι είναι απαραίτητο για ένα οποιοδήποτε από τα 12 Pieces χωρίς να συμπεριλαμβάνει τα ειδικά χαρακτηριστικά που έχουν μερικά από αυτά .

**Το Piece interface περιέχει τις εξής methods :**

1. public int getPower( ); returns the power of the Piece
2. public void setPower( ); sets the Power for the Piece
3. public String getType( ); returns the name of the Piece(Dragon, Mage…)
4. public void setType( ); sets the type of the Piece
5. public int getPlaceInTheBoardX( ); return the axis x coordinates of the Piece
6. public void setPlaceInTheBoardX( ); sets the axis x coordinates of the Piece
7. public int getPlaceInTheBoardY( ); return the axis y coordinates of the Piece
8. public void setPlaceInTheBoardY( ); sets the axis y coordinates of the Piece
9. public boolean getAlive(); returns true if the piece is alive
10. public void setAlive( ); sets the new Piece alive
11. public Color getColor( ); returns the color of the Piece(blue or red)
12. public void setColor( ); sets the color of the Piece(blue or red)
13. public Piece[ ][ ] move(Piece[ ] k, Piece[ ][ ] o , int x, int y, int[ ][ ] greenplaces.int turn) ; Moves the Piece to (x,y)

Στην Β φάση προστέθηκε η μέθοδος move η οποια υπήρχαι στην κλάση MovablePiece.

Παρακάτω φαίνονται οι κλασεις ImovablePiece και MovablePiece οι οποίες υλοποιούν το interface Piece.

**Class ImmovablePiece**

Για αυτήν την κλάση δεν αναφέρω τις μεθόδους που κάνει Override. Η κλάση αυτή αφορά μόνο τα Pieces τα οποία δεν κινούνται (flag, bomb).

Attributes:

1. private int power; The power of the Piece(11 for the bomb, 0 for the flag)
2. private Color col; Color of the Piece
3. private boolean alive; True if it is alive
4. private int placeInTheBoardX; The axis x coordinates of the Piece
5. private int placeInTheBoardY; The axis y coordinates of the Piece
6. private String type; The type of the immovable Piece

Methods:

Η κλαση ImmovablePiece δεν έχει κάποιο παραπάνω method.

**Class MovablePiece**

Για αυτήν την κλάση δεν αναφέρω τις μεθόδους που κάνει Override. Η κλάση αυτή αφορα τα Pieces που κινούνται επίσης κανει implement το interface Piece και extend την κλάση Board .

Attributes:

1. private int power; The power of the Piece
2. private Color col; Color of the Piece
3. private boolean alive; True if it is alive
4. private int placeInTheBoardX(); The axis x coordinates of the Piece
5. private int placeInTheBoardY(); The axis y coordinates of the Piece
6. private String type; The type of the movable Piece
7. private int idOfPiece; The rank of the Piece (10 Dragon, 9 Mage…)
8. private Board b;

Methods:

1. public int Attack(Piece defenderEnemy); This method returns 0 if the attacker won 1 if the defender won 2 if it is a draw.
2. public int getIdOfPiece( ); Returns the id of the Piece (1, 2, 3,...,10).
3. public void setIdOfPiece( ); Sets the id of the Piece (1, 2, 3,...,10).

**Class SpecialMovablePiece**

Η κλάση SpecialMovablePiece κάνει extend την κλάση MovablePiece . Αφορά τα Pieces που έχουν κάποιες παραπάνω δυνάμεις από τα απλά Pieces (Slayer, Scout, Dwarf).Το μονο που περιέχει η κλάση αυτή είναι ένας constractor.

**Classes Dwarf Scout Slayer**

Αυτές οι κλάσεις κάνουν extend την κλάση SpecialMovablePiece και override κάποιες από τις μεθόδους της MovablePiece

Class Dwarf:

Η κλάση Dwarf κάνει override την μέθοδο Attack διότι αν ο Dwarf επιτεθεί σε παγίδα τότε η παγίδα εξολοθρεύεται.

Class Scout:

Η κλάση Scout κάνει override την μέθοδο move διότι ο Scout μπορεί να κινηθεί πάνω από ένα βήμα την φορά.

Class Slayer:

Η κλάση Slayer κάνει override την μέθοδο Attack επειδή αν επιτεθεί στον Dragon παρόλο που ο Slayer έχει λιγότερο power νικάει τον Dragon.

**Classes that extends MovablePiece and ImmovablePiece**

* **Movable Pieces Classes**
* Dragon
* Mage
* Knight
* Beast Rider
* Sorceress
* Lava Beast / Yeti
* Elf
* **Immovable Pieces Classes**
* Flag
* Bomb

Αυτό το κεφάλαιο αναφέρεται στα πιόνια του παιχνιδιού (τις παγίδες ,την σημαία και ολα τα “movable pieces”). Ουσιαστικά δίνουμε τις τιμές του id του piece ,το type ,το power και το isAlive .

**Class Board**

Η κλάση αυτή αναπαρηστά το ταμπλό στο οποίο εκτελείται το παιχνίδι.

Attributes:

1. private Piece[ ][ ] board; It is a 8 x 10 board that has all the pieces that are alive from each player.

Methods:

1. public Piece[ ] [ ] getBoard(); It returns the board.
2. public void setBoard(); It sets the board and makes changes when a Piece is moving
3. public Piece[ ] [ ] initialize( ); It initializes the board with all the Pieces on each side and the yellow areas.
4. public int [ ] [ ] greenPlaces( ); It returns an 8 x 10 array that has the number 1 in every place that the Piece can move.

Στην Β φάση δεν άλλαξε τιποτα στο Board.

**Class Player**

Αυτή η κλάση αναπαριστά τους παίκτες οι οποίοι διαχειρίζονται τα Pieces.

Attributes:

1. private ArrayList<Piece> piecesAlive; This array list sees the pieces that are alive.
2. private ArrayList<Piece> piecesCaptured; This array list sees the pieces that are captured .
3. private String name; This is the name of the player.
4. private double winPercentage; It represents the percentage of successful attacks.
5. private int revives; This is the amount of the revives that the player has done.

Methods:

1. public ArrayList<Piece> initializeTheRedList( ); It makes a red army.
2. public ArrayList<Piece> initializeTheBlueList( ); It makes a Blue army.
3. public ArrayList<Piece> getPiecesAlive( );Returns the Pieces that are alive.
4. public ArrayList<Piece> setPiecesAlive(Piece deadPiece); Removes the dead Pieces.
5. public ArrayList<Piece> getPiecesCaptured( ); It returns the captured Pieces.
6. public ArrayList<Piece> setPiecesCaptures(Piece capturedPiece); Adds the captured Piece.
7. public double getWinPercentage( ); Returns the win percentage of the player.
8. public void setWinPercentage( ); Set the win Percentage.
9. public int getRevives( ); Returns the number of the revives that are done.
10. public void setRevives( ); Sets the number of the revives.

Στην κλάση Player δεν άλλαξε τιποτα.

## **Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller**

Ο Controller είναι υπεύθυνος για να ξεκινήσει το παιχνίδι , δηλαδή να αρχικοποιήσει το Board , να δημιουργήσει τους δύο παίκτες και να αρχικοποιήσει τις λίστες με τα πιόνια τους , να ελέγχει ποιός παίζει , να κάνει τις αλλαγές στο board και να τερματίζει το παιχνίδι όταν πρέπει. Ουσιαστικά ο Controller διαχειρίζεται ολα τα μοντέλα (Pieces, Board, Player…) του παιχνιδιού με βάση τα δεδομένα που λαμβάνει από το View.

**Class Controller**

Attributes:

1. private Player player1, player2; These two objects are the two players that are playing the game.
2. private Board board; This is the board where the game takes place.
3. private Piece[ ][ ] pieces; In this array we put the pieces and the yellow areas.
4. private int turn; This is how we check who’s player turn is.

Methods:

1. public Board changeBoard(); This method changes the board when a Piece moves.
2. public int changeturn( );It adds 1 to the turn every time .
3. public terminateGame( ); Return true if the game has to be terminated.
4. getTurn( ); Returns the turn.
5. setTurn(int add ); Sets the turn.
6. copieBoard(Piece[ ][ ] board ); copies the board that is given.

Στην Β φάση προστέθηκαν οι συναρτήσεις getTurn, SetTurn, copieBoard.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View είναι υπεύθυνο για το τι βλέπει ο χρήστης και για την αλληλεπίδραση του με το παιχνίδι. Συγκεκριμένα το View περιέχει τα 60 κουμπιά που είναι όλα τα Pieces και τα τοποθετεί στο board. Ακόμα δημιουργεί το menu δίπλα από το ταμπλό, τοποθετεί την εικόνα που αντιστοιχεί στο κάθε Piece στο κουμπί του και τελος κάθε φορά που ο παίκτης κάνει revive εμφανίζεται ένα πλαίσιο που του επιτρέπει να διαλέξει ποιό Piece θέλει να κάνει revive.

**Class GraphicUi**

Contractor:

Ουσιαστικά ο Constructor δημιουργεί το παράθυρο και τοποθετεί τα πιόνια στο Board.

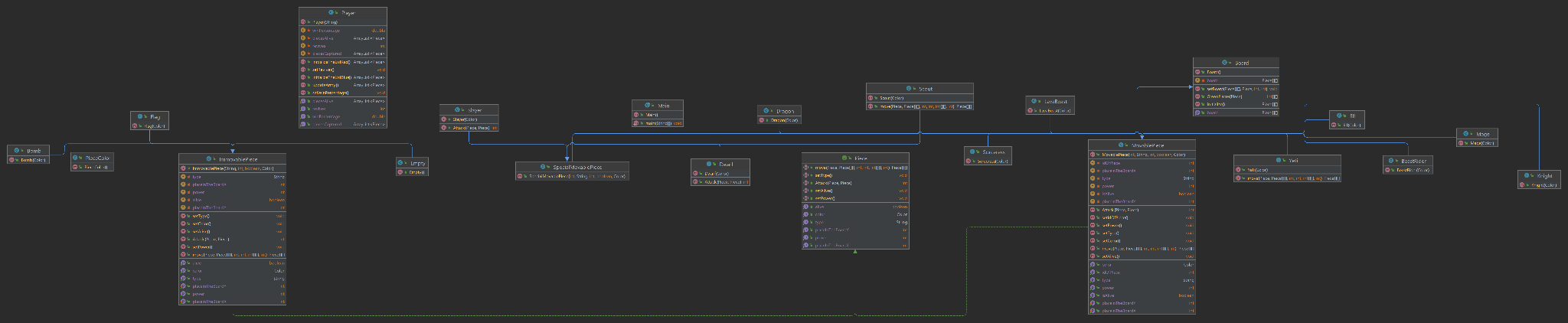
Methods:

1. public Image resizeza(Image img); Resizes the image that is given.
2. public Piece placeToMove(int x, int y, Piece[ ][ ] b); Returns the Piece that the user wants to move.

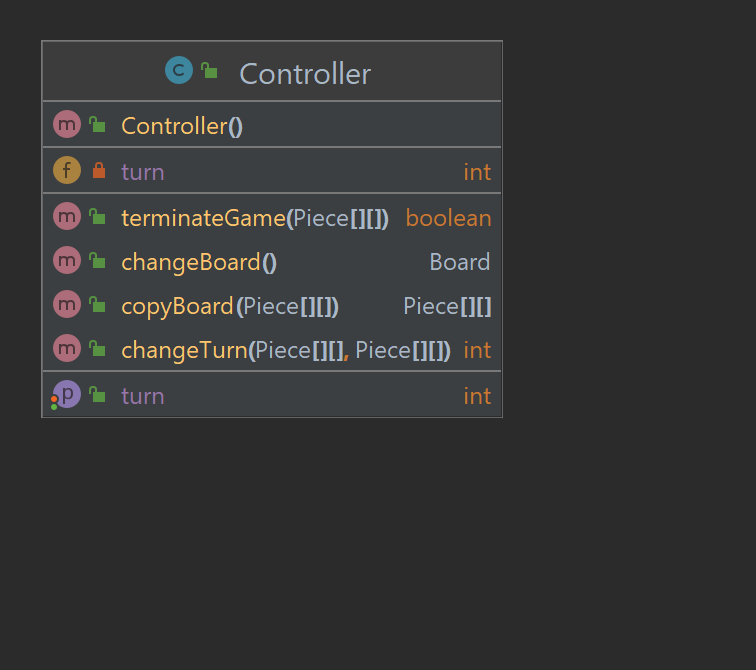
Στην φάση B προστέθηκαν οι συναρτήσεις resizeza και placeToMove .

## **Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML**

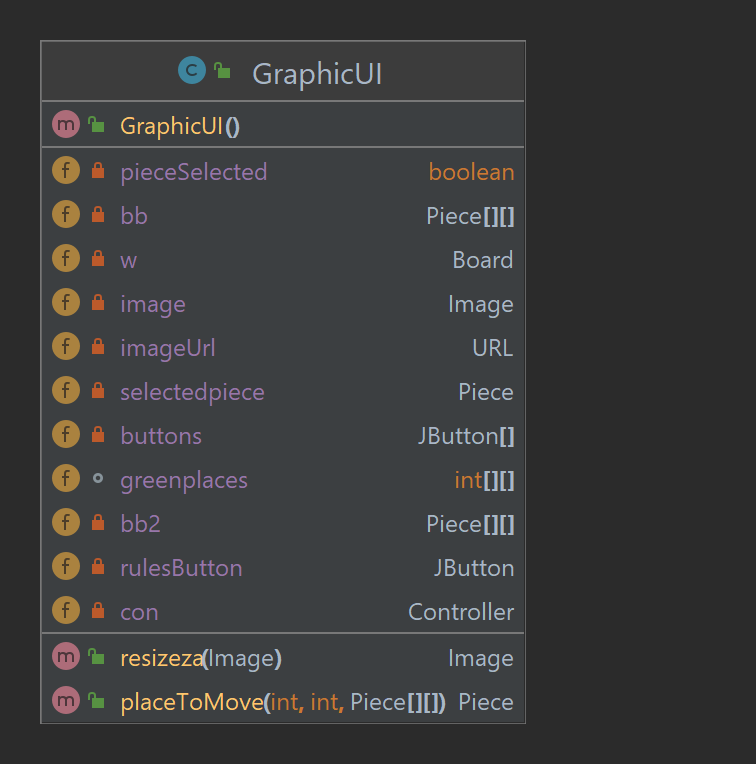
* Στην παρακάτω εικόνα φαίνεται το package model( Υπαρχει και σε ξεχωριστη εικόνα στο αρχείο).



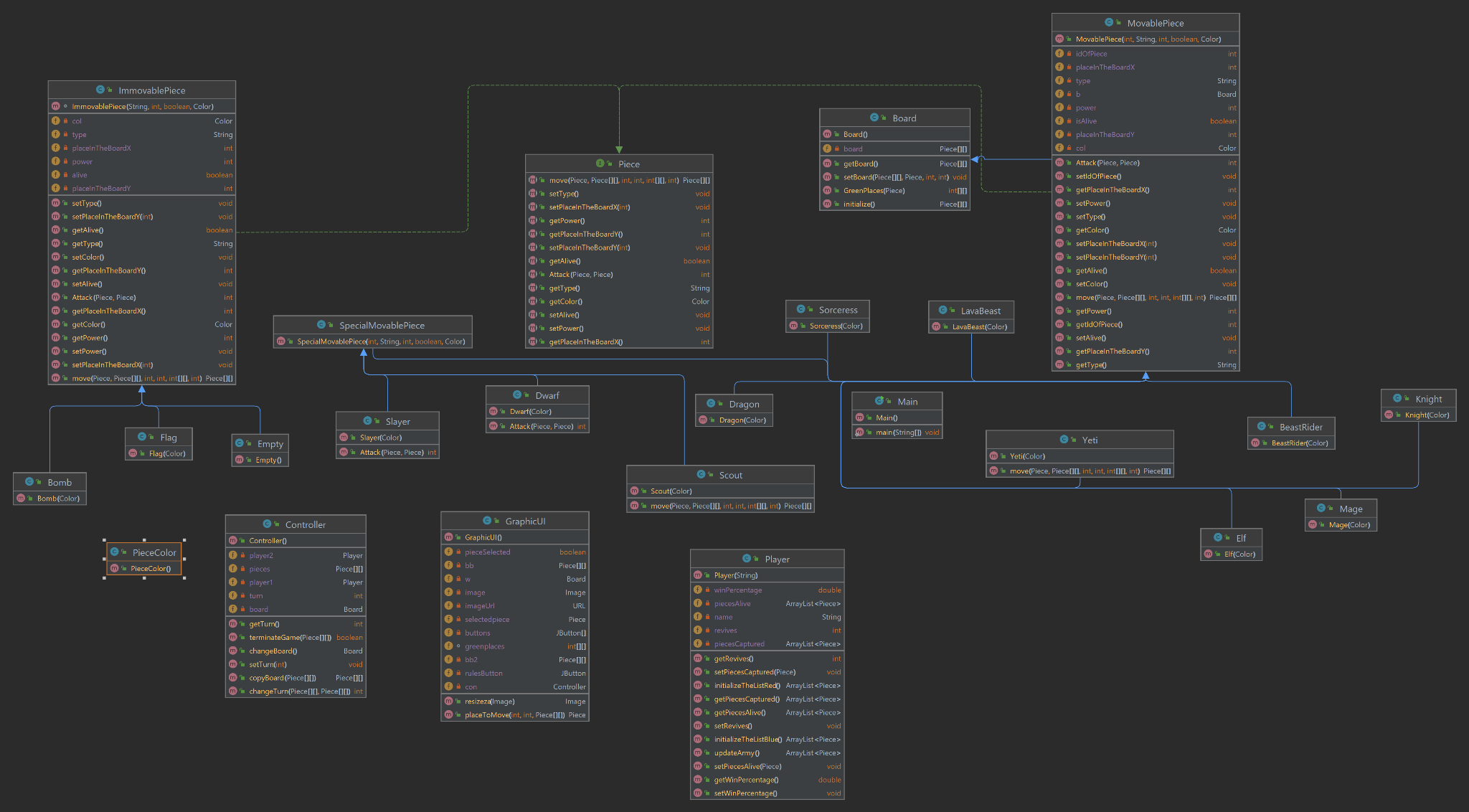
* Στην παρακάτω εικόνα φαίνεται το package Controller.



* Στην παρακάτω εικόνα φαίνεται το package View.



* Στην παρακάτω εικόνα φαίνονται όλα τα UML.



1. **Λειτουργικότητα**

**Τα ερωτήματα που υλοποιήθηκαν είναι τα εξείς :**

1. Τήρηση σειράς και τοποθέτηση πιονιών
2. Σωστή Κίνηση πιονιών και ανιχνευτή
3. Επίθεση πιονιών
4. Επίθεση εξολοθρευτή-δράκου
5. Εξουδετέρωση παγίδας από νάνο
6. Τέλος παιχνιδιού-κατάκτηση σημαίας

**Τα ερωτήματα που δεν υλοποιήθηκαν είναι τα εξείς :**

1. Διάσωση πιονιού που βγήκε από το παιχνίδι
2. Μενού πληροφοριών στο πλάι
3. Game mode: Μειωμένος Στρατός
4. **Συμπεράσματα**

Η εργασία είχε αρκετό ενδιαφέρον και πειραματισμό με την java και το GUI. Μου έδωσε την δυνατότητα να ασχοληθώ με project management και γραφικό περιβάλλον για πρώτη φορά όπως και με την δημιουργία ενός επιτραπέζιου παιχνιδιού στην java. Δυστυχως δεν κατάφερα να υλοποιήσω το menu , το game mode μειωμένος στρατος και το revive λογο περιορισμένου χρόνου . Παρόλα αυτά η διαδικασία της δημιουργίας του παιχνιδιού ήταν πολύ ευχάριστη και δημιουργική.