

OBJECT-ORIENTED PROGRAMMING HY-252

PROJECT 2015-2016

Designed by Nikolaos Simidalas



Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller). Ο Controller θα είναι ο συνδετικός κρίκος των Model και View.

Η συνολική εργασία αποτελείται από τα εξής πακέτα:

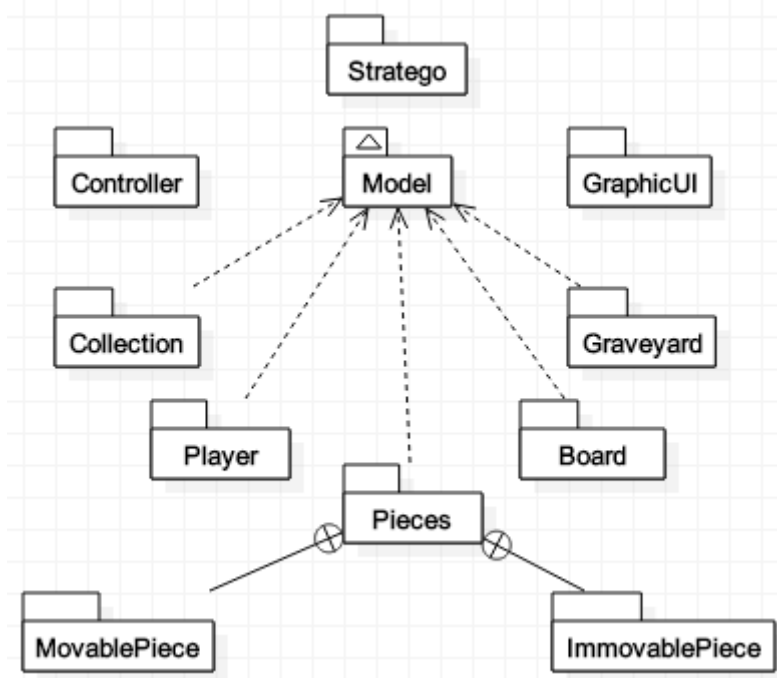
Package Stratego: Περιέχει την κλάση main μέσω της οποίας τρέχει το παιχνίδι.

Package Controller: Συνδέει την γραφική απεικόνιση με το μοντέλο του παιχνιδιού.

Package GraphicUI: Υλοποιεί την γραφική απεικόνιση.

Package Model: Περιέχει όλα τα συστατικά του παιχνιδιού.

- Package Collection:
- Package Graveyard:
- Package Board:
- Package Player:
- Package Pieces: Περιέχει τις κλάσεις των διάφορων πιονιών του παιχνιδιού
 - Package MovablePiece
 - Package ImmovablePiece



Package Model

Σε αυτό το πακέτο θα παρέχεται η abstract class Piece, οι υποκλάσεις της (ImmovablePiece & MovablePiece) καθώς επίσης και κλάσεις που κληρονομούν τις δύο προηγούμενες. Δηλαδή κάθε κλάση - χαρακτήρας του παιχνιδιού.

Τέλος, το πακέτο αυτό θα παρέχει τις κλάσεις Graveyard, Collection, Board & Player.

Piece

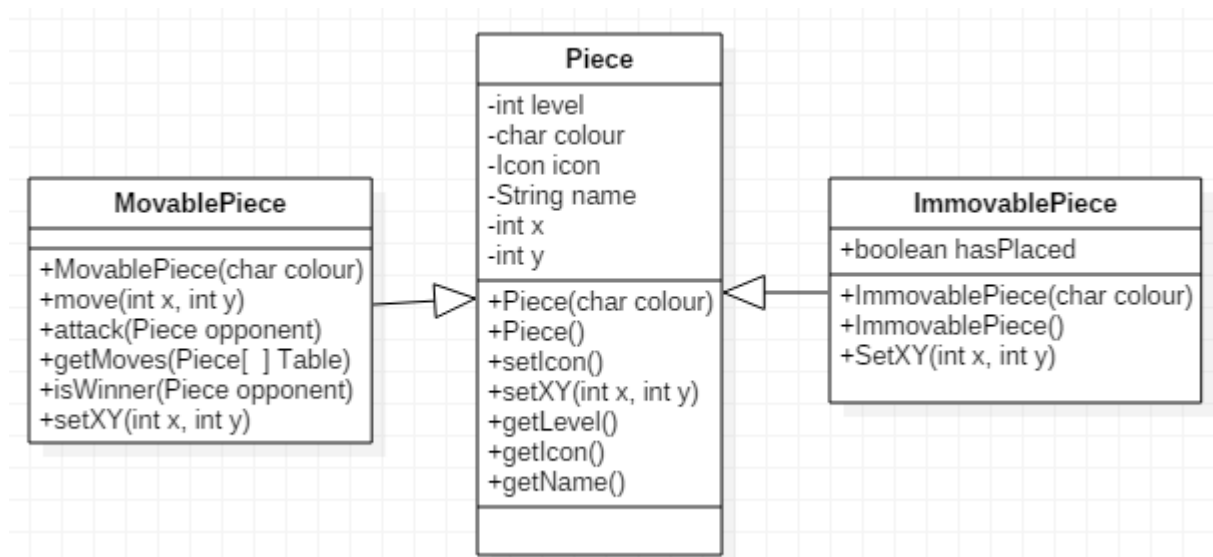
Η κλάση αυτή περιέχει κάποιες γενικές μεθόδους & attributes τα οποία θα χρησιμοποιηθούν με τον ίδιο τρόπο από κάθε άλλη υποκλάση της.

THE ATTRIBUTES:

1. private int level; // The level of a piece (From 0 to 10)
2. private char colour; // The colour of a piece(R(Red) or B(Blue))
3. private Icon icon; // The image of a card(set or flip)
4. private String name; // The name of a card(Dragon, Mage, Knight, ...)
5. private int x; // X coordinate
6. private int y; // Y coordinate

THE METHODS:

1. public void setIcon() ; Transformer(mutative)
⇒ Sets the icon of this piece.
2. public int getLevel() Accessor
⇒ Returns the level of this piece.
3. public Icon getIcon() Accessor
⇒ Returns the icon of this piece.
4. public String getName() Accessor
⇒ Returns the name of this piece.
5. public abstract void setXY(int x, int y); Transformer(mutative)
➔ Abstract class which will be implemented from the subclasses
⇒ Sets the coordinates of the piece.
6. public int getX(); Accessor
⇒ Returns the x coord.
7. public int getY(); Accessor
⇒ Returns the y coord.



Στην συνέχεια έχουμε τις abstract κλάσεις MovablePiece & ImmovablePiece που υλοποιούν την Piece.

MovablePiece

Η κλάση αυτή περιέχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους απ' αυτές που περιέχει η superclass κλάση piece καθώς επίσης, κάνει overriding την μέθοδο getXY();

THE ATTRIBUTES:

1. private boolean hasRecovered;

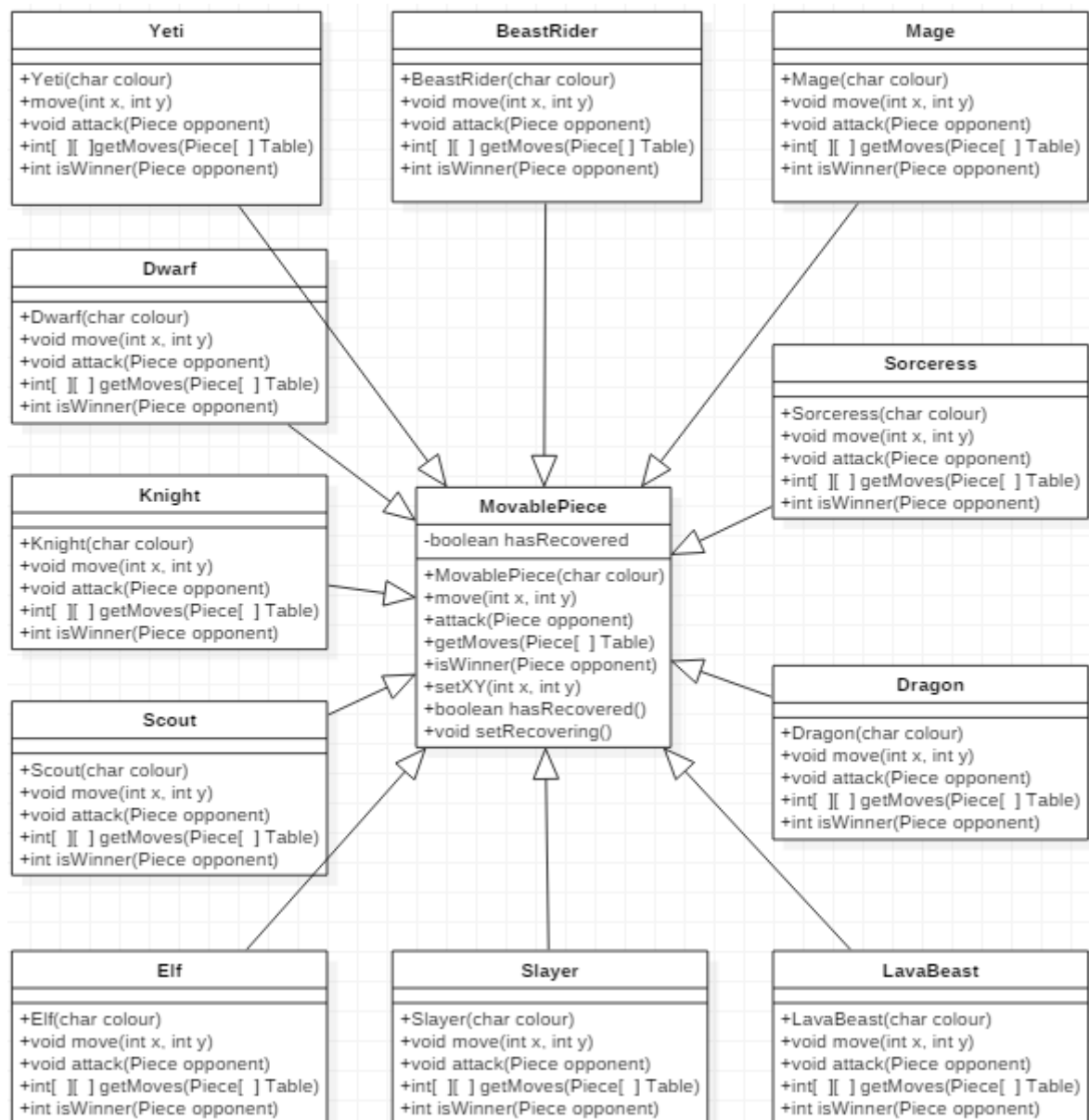
THE METHODS:

1. abstract public void move(); Observer
 ➔ Abstract method which will be implemented different in every piece.
 Returns true if there is at least one moving.
2. abstract public void attack(Piece opponent); Transformer(mutative)
 ➔ Abstract method which will be implemented different for every piece.
 ⇨ A to an opponent piece.
3. abstract public int[] getMoves(int x, int y); Accessor
 ➔ Abstract method which will be implemented different for every piece.
 ⇨ Return an array with the possible movings.

4. `abstract public int isWinner(Piece opponent) ; Observer`
 - ➔ Abstract method which will be implemented different in some pieces.
 - ⇒ In general case returns 1 if this piece's level is higher than the opponent's.
5. `public void setXY(int x, int y); Transformer(mutative)`
 - ⇒ Set the coordinates of the piece.
6. `public boolean hasRecovered(); Observer`
 - ⇒ Returns true if has recovered a piece before
7. `public void setRecovering();Transformer(mutative)`
 - ⇒ Make variable hasRecovered equal to true

Την abstract κλάση Movable την κληρονομούν οι παρακάτω 11 κλάσεις:

1. Yeti
2. BeastRider
3. Mage
4. Dwarf
5. Sorceress
6. Knight
7. Scout
8. Dragon
9. Elf
10. Slayer
11. LavaBeast



ImmovablePiece

Η κλάση αυτή περιέχει ένα επιπλέον attribute απ' αυτά που περιέχει η superclass κλάση piece καθώς επίσης, κάνει overriding την μέθοδο setXY();

THE ATTRIBUTES:

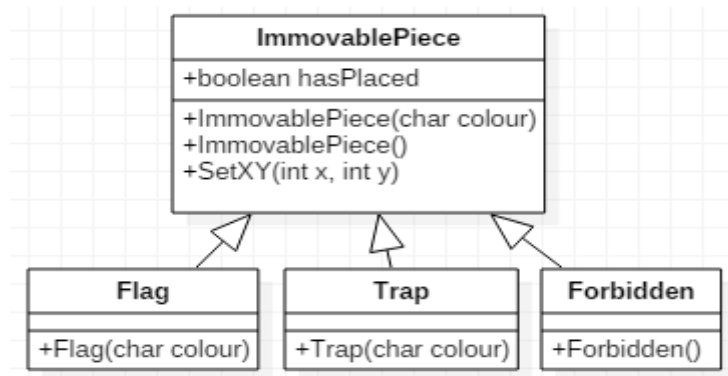
1. private boolean hasPlaced; // Only once is able to be places somewhere.

THE METHODS:

1. public void setXY(int x, int y); Transformer(mutative)
⇒ Set the coordinates of the piece.

Την abstract κλάση Immovable την κληρονομούν οι παρακάτω 3 κλάσεις:

- 1) Flag
- 2) Trap
- 3) Forbidden // It's a space on the table which none piece can use.



Graveyard

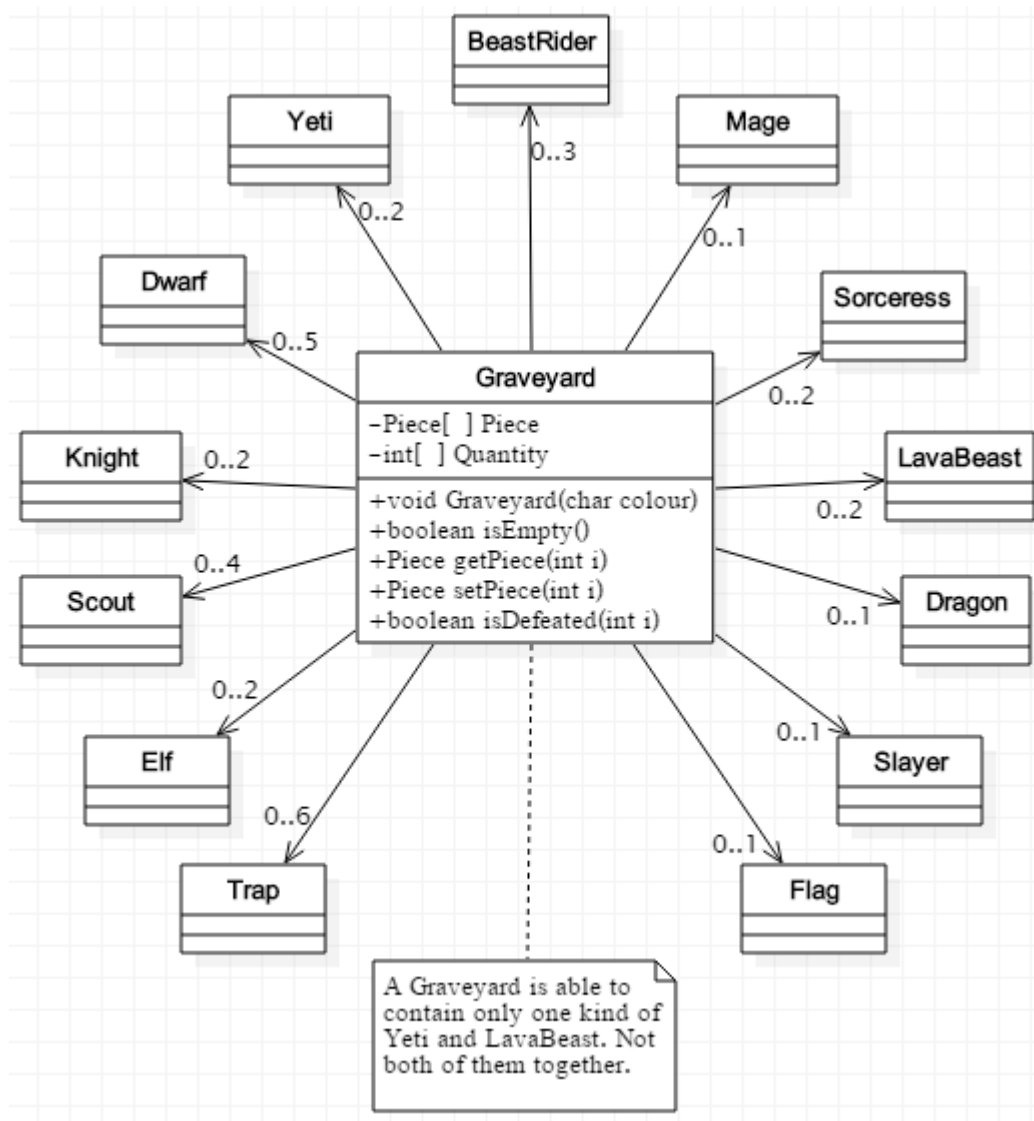
Περιέχει όλα τα πόνια ενός παίκτη τα οποία έχουν βγει από το παιχνίδι. Χρησιμοποιείται, επίσης, όταν θέλουμε να επαναφέρουμε κάποιο πόνι στο παιχνίδι μέσω της ιδιότητας κάποιου άλλου πιονιού.

THE ATTRIBUTES:

1. `private Piece[] Piece` // Contains the type of the piece which exist
2. `Private int[] Quantity;` // Contains the quantity of every piece.

THE METHODS:

1. `public boolean isEmpty(); Observer`
 ⇒ Returns true if this list contains no elements.
2. `public boolean isDefeated(int i); Observer`
 ⇒ returns true if there is at least a defaulted piece
3. `public Piece getPiece(int i); Accessor`
 ⇒ returns the piece in position i
4. `public void setPiece(int i); Transformer`
 ⇒ Rise the quantity of piece i



Collection

Περιέχει όλα τα πιόνια ενός παίκτη τα οποία είναι προς εισαγωγή.
Χρησιμοποιείται μόνο στην αρχή του παιχνιδιού, όπου κάθε παίκτης στήνει τα πιόνια του σύμφωνα με την εκάστοτε στρατηγική.

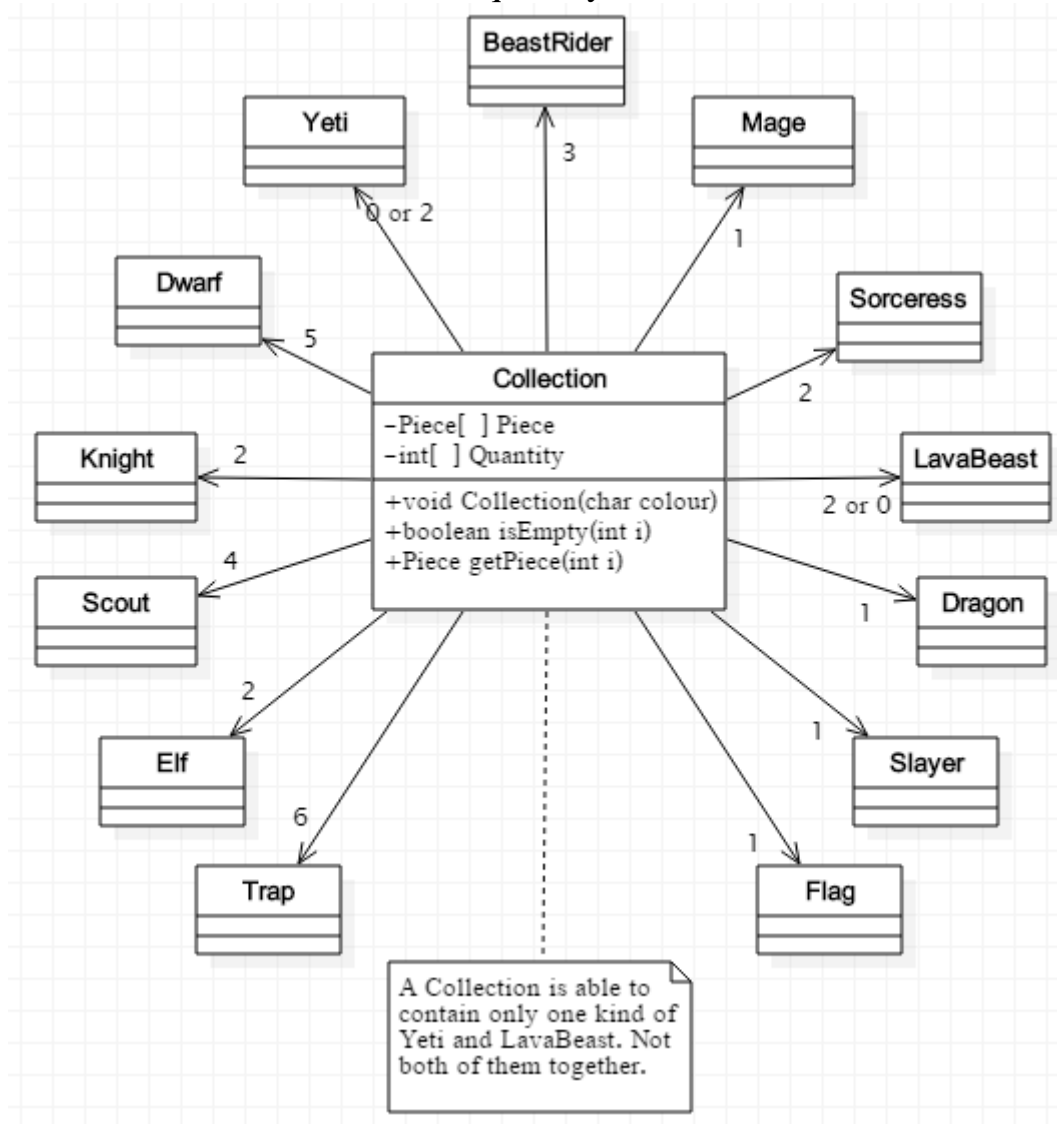
THE ATTRIBUTES:

1. private Piece[] Piece // Contains the type of the piece which exist
2. Private int[] Quantity; // Contains the quantity of every piece.

THE METHODS:

1. public Piece getPiece(int i); Accessor
⇒ returns the piece in position i.

2. public boolean isEmpty(int i); Observer
 ⇒ Return true if there is quantity, or false different.



Player

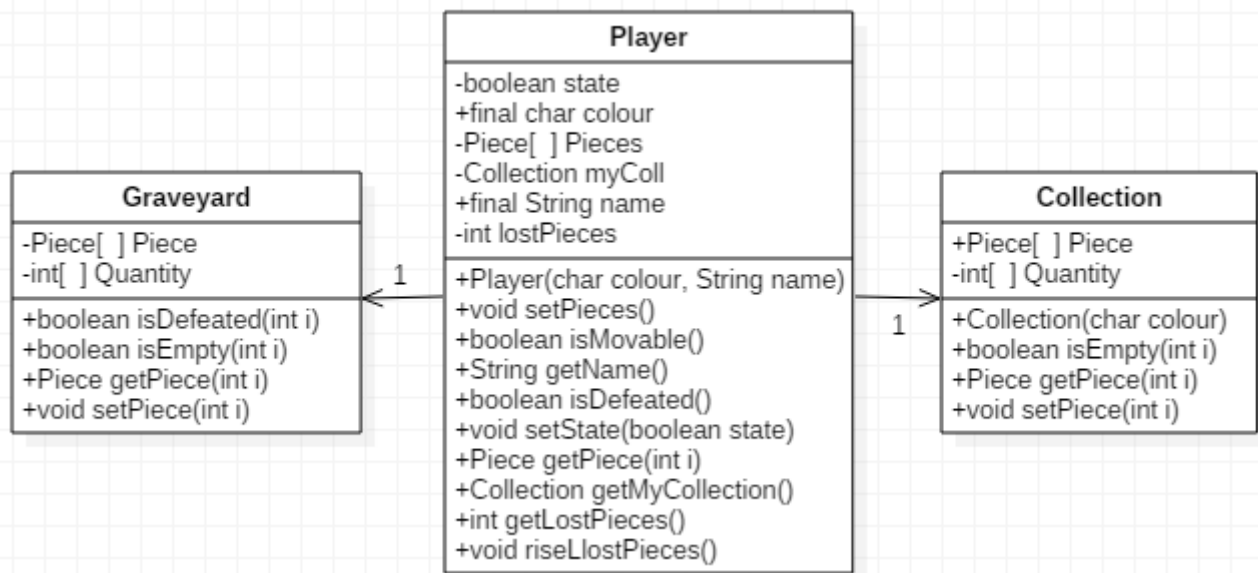
Η κλάση αυτή αναπαριστά κάθε παίκτη του παιχνιδιού.
 Συνδέεται τόσο με ένα Collection όσο και με ένα Graveyard.

THE ATTRIBUTES:

- | | |
|-------------------------------------|---------------------------------------|
| 1. private boolean state; | // Is or not defeated |
| 2. final private char colour; | // What colour player is |
| 3. private Piece[] Pieces; | // Contains its pieces |
| 4. final private Collection myColl; | // Standard collection with pieces |
| 5. private Graveyard myGrave; | // A graveyard for reviving of pieces |
| 6. final private String name; | // What its name |
| 7. private int lostPieces; | // How many piece has lost |

THE METHODS:

1. `public void setPieces(); transformer(mutative)`
⇒ Sets the pieces of a player for a new game
2. `public boolean isMovable(); Observer`
⇒ Returns true if the player can make a moving else false
3. `public String getName(); Accessor`
⇒ Returns name of the player
4. `public boolean isDefeated(); Observer`
⇒ Returns true the player has lost all it's pieces
5. `public void setState(); transformer(mutative)`
⇒ sets the state of the player.
6. `public Piece getPiece(int i); Accessor`
⇒ returns the piece in position i
7. `public Collection getMyCollection(); transformer(mutative)`
⇒ returns a collection with all the pieces of the player
8. `public int getLostPieces(); Observer`
⇒ Returns the number of the defeated pieces
9. `public void riseLostPieces(); transformer(mutative)`
⇒ sets 1 plus lost piece
10. `public Graveyard getMyGraveyard(); Accessor`
⇒ Returns a Graveyard with all the pieces of the player.



Board

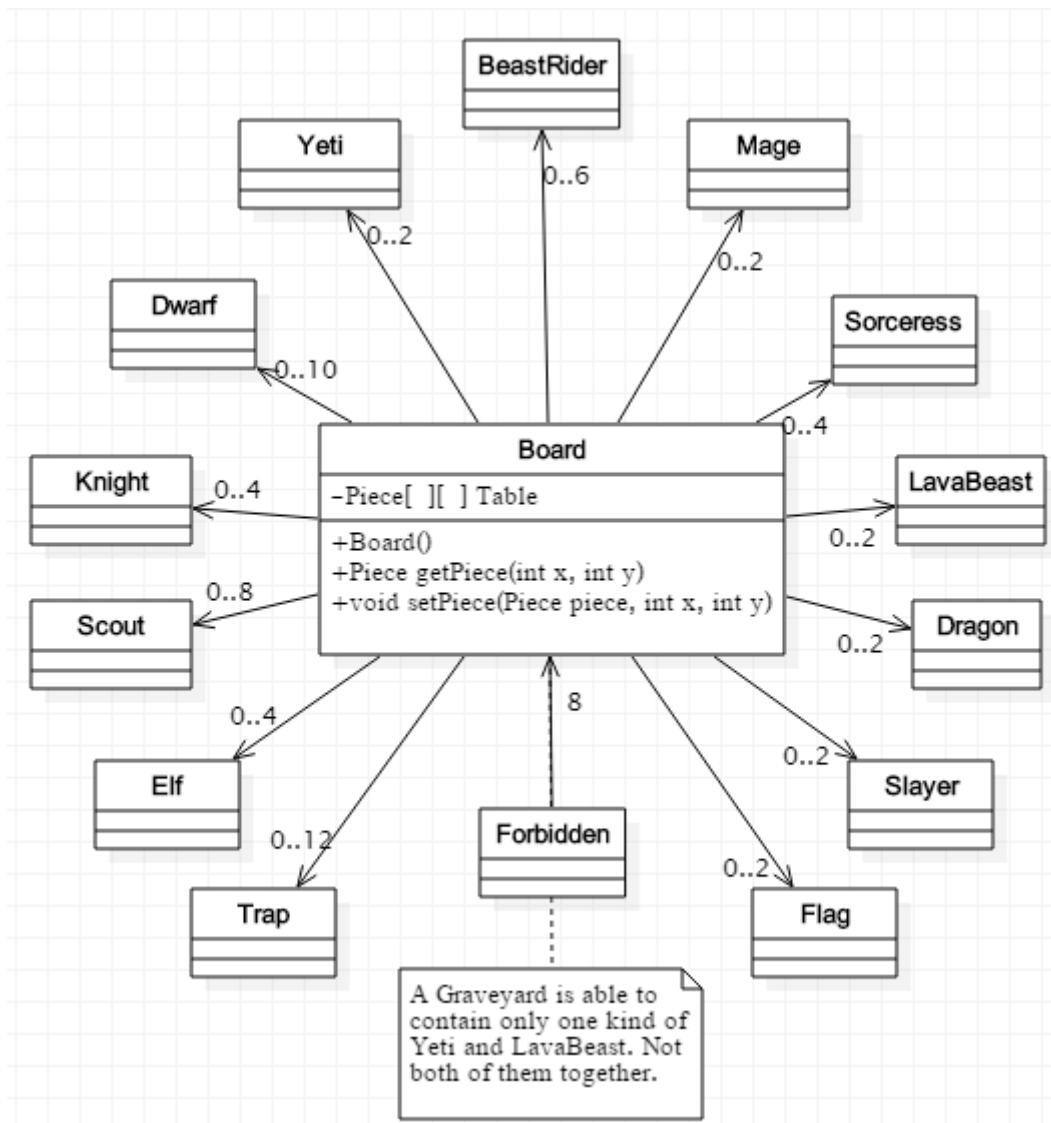
Η κλάση αυτή αναπαριστά το ταμπλό του παιχνιδιού.
Επίσης, μας παρέχει μεθόδους για την τροποποίηση του.

THE ATTRIBUTES:

- 1) private Piece[][] Tale // Contains pieces in suitable positions

THE METHODS:

- 1) public Piece getPiece(int x, int y) Accessor
⇒ returns the piece in position (x, y)
- 2) public void setPieces(Piece piece, int x, int y); transformer(mutative)
⇒ Set a piece on the table



Package Controller

Το πακέτο αυτό συνδέει το μοντέλο του παιχνιδιού με την γραφική απεικόνιση. Πιο συγκεκριμένα, λαμβάνει ως είσοδο τις ενέργειες που θέλουμε να εκτελεστούν πατώντας κάποιο κουμπί και παράγει ως έξοδο την επανασχεδίαση του ταμπλό του παιχνιδιού, κάνοντας κατάλληλες επεξεργασίες τόσο σε αυτό, όσο και στα επιμέρους μέρη που το αποτελούν.

Controller

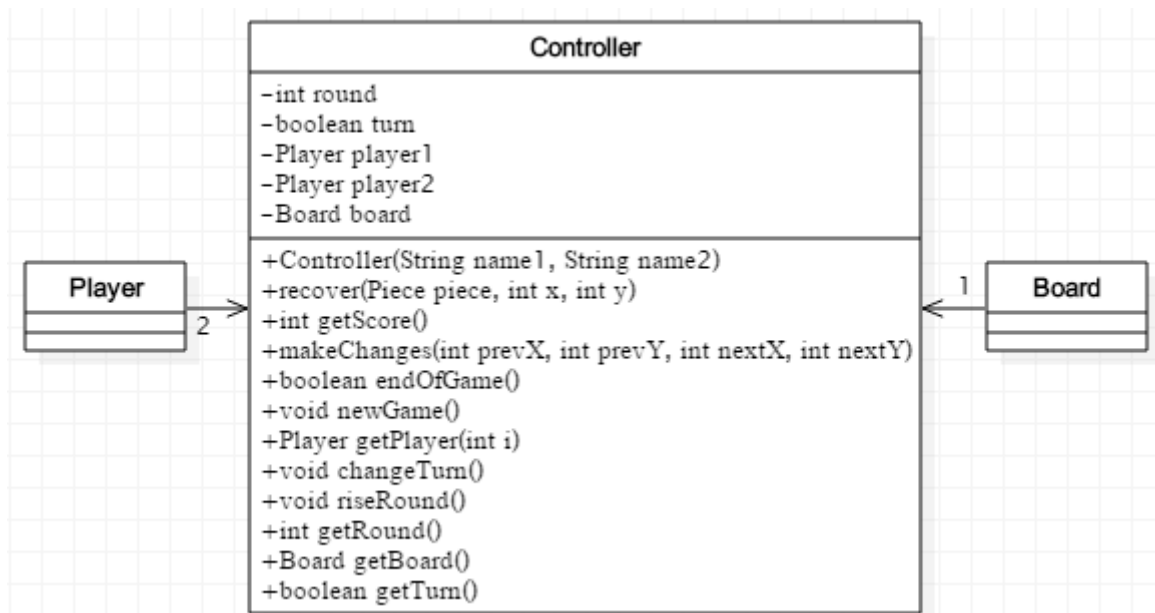
Αυτή είναι η μια και μοναδική κλάση του package Controller η οποία κάνει όλη τη δουλειά του παιχνιδιού. Ο ρόλος της είναι να παίρνει τις επιλογές του παίκτη μέσω της γραφικής απεικόνισης και να εκτελεί την κατάλληλη λειτουργία πάνω στα συστατικά του παιχνιδιού, δηλαδή το model. Μερικές άλλες λειτουργίες της είναι η έναρξη ενός νέου παιχνιδιού, ο τερματισμός αυτού και η διατήρηση του σταδίου όπου βρίσκεται το παιχνίδι. Περιέχει τα παρακάτω attributes & methods.

THE ATTRIBUTES:

1. private int round; // Hold the round which has passed
2. private boolean turn; // Say who player's turn is
3. private Player player1, player2; // The players of the game
4. private Board board; // The table of the game

THE METHODS:

1. public void recover(Piece piece, int x, int y); transformer (mutative)
⇒ Recovers a piece
2. public int getScore(); Accessor
⇒ return how is the difference between the lost pieces of the players
3. public void makeChanges(int prevX, int prevY,int nextX, int nextY); Transf.
⇒ Swap the position of a piece end more others
4. public boolean endOfGame(); Observer
⇒ Return true if a game has finished.
5. public void newGame(); Transformer(mutative)
⇒ Construct a new game
6. public Player getPlayer(int i); Accessor(selector)
⇒ Return a the player i
7. public void changeTurn(); Transformer(mutative)
⇒ Change the turn of the game
8. public void riseRound(); Transformer(mutative)
⇒ Rise the number of the rounds of the game which have passed
9. public int getRound();Accessor(selector)
⇒ Return the number of the rounds which have passed
10. public Board getBoard();Accessor(selector)
⇒ Return the table of the game
11. public boolean getTurn(); Accessor(selector)
⇒ Return who player's turn is



Package View

Το πακέτο αυτό περιλαμβάνει την κλάση `GraphicUI` που αναπαριστά το ταμπλό του παιχνιδιού ως γραφική απεικόνιση καθώς επίσης περιλαμβάνει και μερικούς `action listener` οι οποίοι υλοποιούν τις ενέργειες που επιθυμεί ο παίκτης να εκτελεστούν.

Πιο συγκεκριμένα, εμπεριέχει ένα ταμπλό 10 X 12. Τα 30 αρχικά απ' αυτά ανήκουν στον έναν παίκτη και άλλα 30 στον άλλο. Περιέχει επίσης κάποιες ελεύθερες θέσεις κίνησης(12), κάποιες απαγορευμένες(8) και κάποιες κρυμμένες, οι οποίες θα βοηθήσουν στην πιο ομαλή εκτέλεση των κινήσεων(36). Στην γραφική απεικόνιση δεν θα εμφανίζονται οι κρυμμένες θέσεις παρά μόνο οι υπόλοιπες.

Επίσης, θα υπάρχουν 3 πρωταρχικά κουμπιά τέτοια ώστε να γίνεται η επιλογή τοποθέτησης των πιονίων των δύο παικτών. Αυτά είναι, ένα για τοποθέτηση των πιονίων και των δύο παικτών σύμφωνα με τις επιλογές του παίκτη, ένα για τοποθέτηση μόνο του ενός παίκτη κι ένα για τυχαία τοποθέτηση των πιονίων.

Για την πρώτη και δεύτερη περίπτωση θα υπάρχει ένα κουμπί το οποίο θα πατιέται όταν ο/οι παίκτης/ες είναι έτοιμος/οι να ξεκινήσουν το παιχνίδι. Επιπλέον, το ίδιο κουμπί θα χρησιμοποιηθεί για την επαναφορά ενός πιονιού στο παιχνίδι. ; Τέλος, όλα τα κουμπιά, θα προστίθενται σε ένα `panel`. Ανάλογα με το σε πιο στάδιο βρίσκεται το παιχνίδι θα τοποθετούνται τα κατάλληλα, και στις κατάλληλες θέσεις, κουμπιά.