

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.

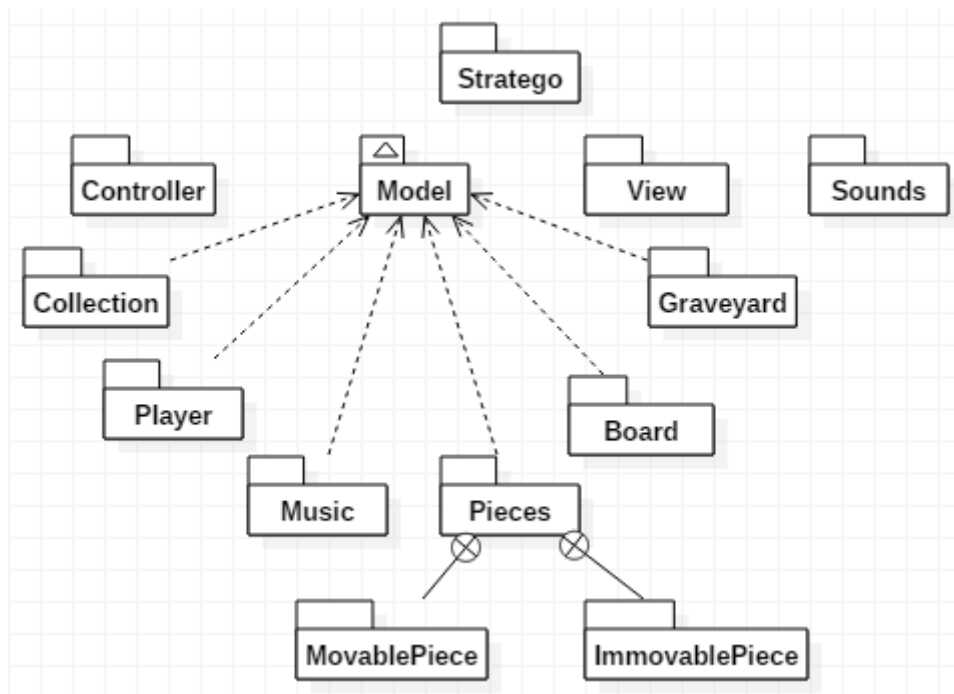
Package Controller: Συνδέει την γραφική απεικόνιση με το μοντέλο του παιχνιδιού.

Package Sounds: Περιέχει τους ήχους που χρησιμοποιούνται στο παιχνίδι.

Package GraphicUI: Υλοποιεί την γραφική απεικόνιση.

Package Model: Περιέχει όλα τα συστατικά του παιχνιδιού.

- Package Collection:
- Package Graveyard:
- Package Board:
- Package Player:
- Package Music:
- 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 & Music και έναν φάκελο με τους ήχους του παιχνιδιού.

Piece

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

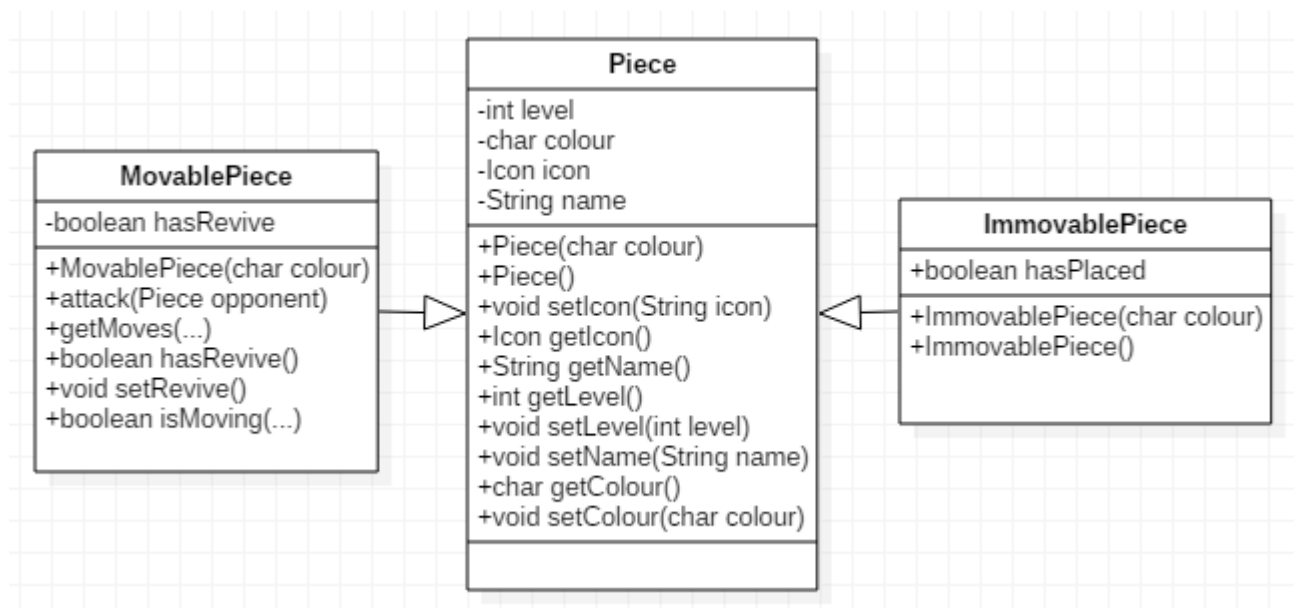
THE ATTRIBUTES:

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, ...)

THE METHODS:

1. public void setIcon(String icon) ; 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 void setColour(char colour);` Transformer(mutative)
⇒ Sets the colour of this piece
6. `public void setLevel(int level);` Transformer(mutative)
⇒ Sets the level of this piece
7. `public void setName(String name);` Transformer(mutative)
⇒ Sets the name of this piece
8. `public char getColour();` Accessor
⇒ Returns the colour of this piece



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:

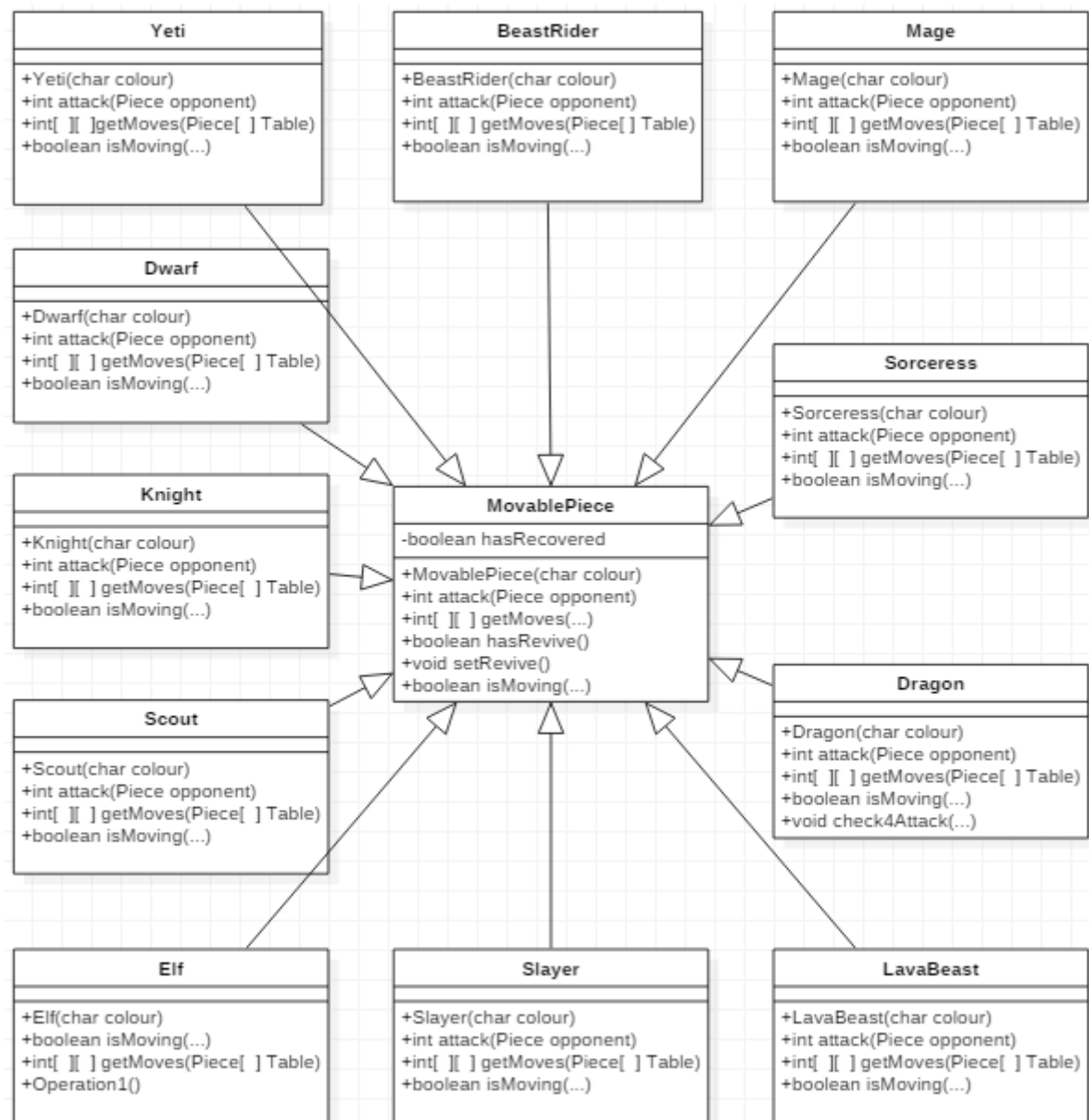
1. private boolean hasRevive; // Keep if the piece has revive another one

THE METHODS:

1. public boolean hasRevive(); Observer
⇒ Check if the piece has revive another one
2. abstract public int attack(Piece opponent); Transformer(mutative)
➔ Abstract method which will be implemented different for every piece.
⇒ Return the type of the attack
3. abstract public int[] getMoves(Piece[] Table, int x, int y); Accessor
➔ Abstract method which will be implemented different for every piece.
⇒ Return an array with the possible movings
4. public boolean isMoving(Piece[] Table, int x, int y); Observer
⇒ Check if the piece is able to move
5. public void setRevive();Transformer(mutative)
⇒ Sets that has revive a piece

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

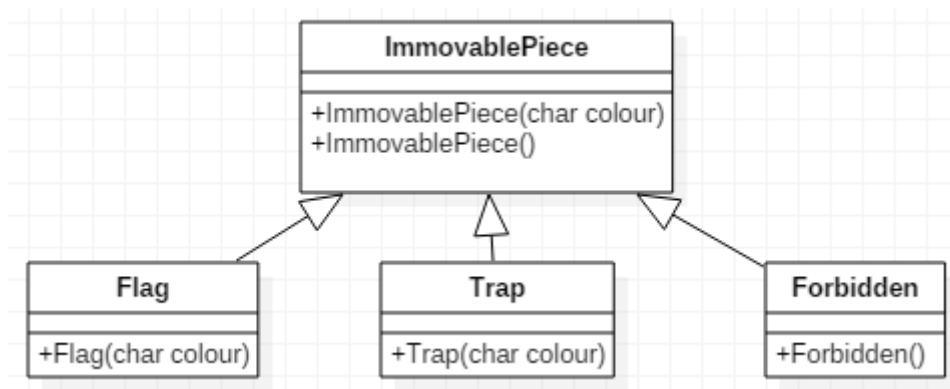


ImmovablePiece

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

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

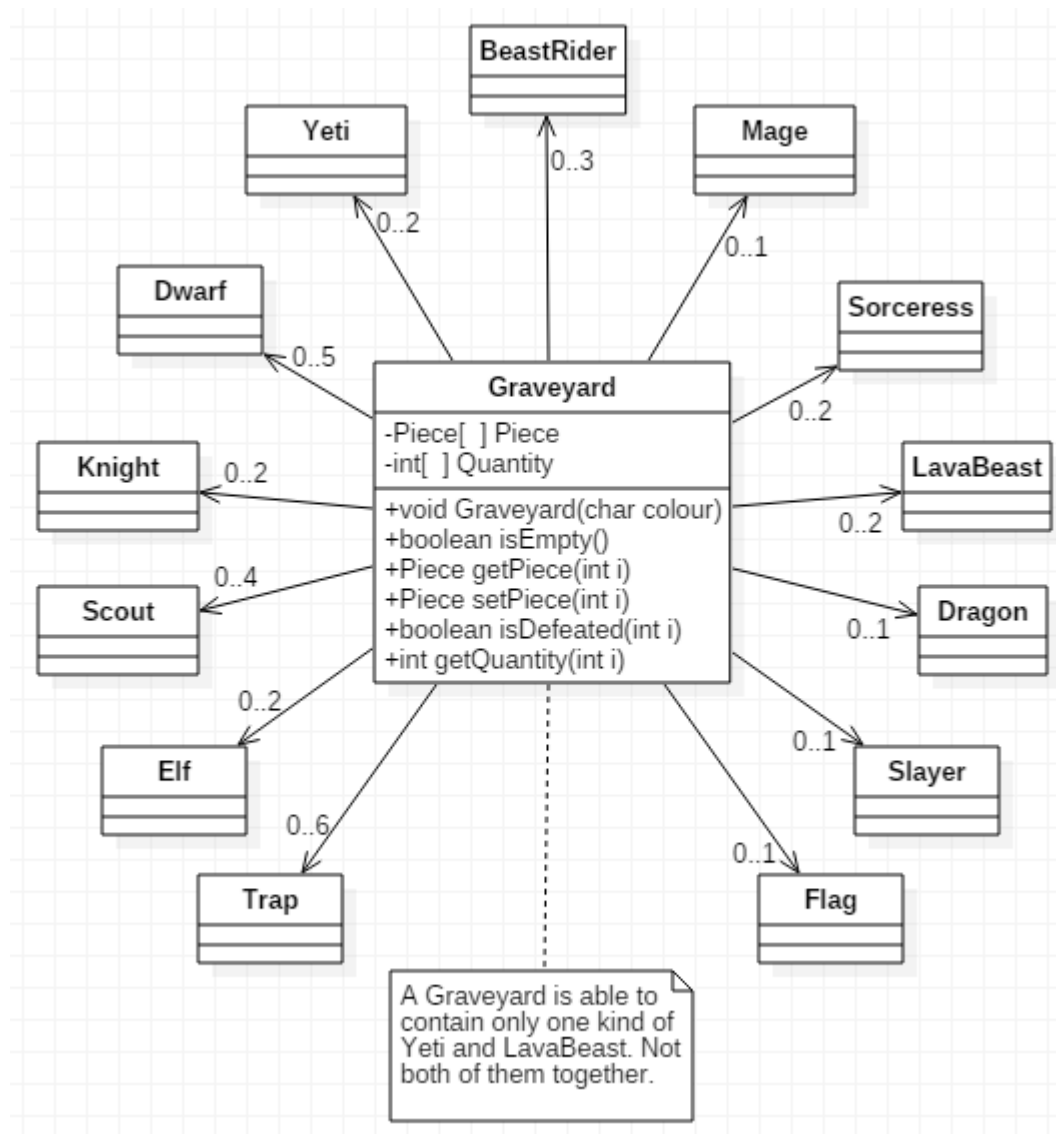
Περιέχει όλα τα πόνια ενός παίκτη τα οποία έχουν βγει από το παιχνίδι. Χρησιμοποιείται, επίσης, όταν θέλουμε να επαναφέρουμε κάποιο πόνι στο παιχνίδι μέσω της ιδιότητας κάποιου άλλου πιονιού.

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(Piece piece); Transformer`
⇒ Rise the quantity of piece piece
5. `public int getQuantity(int i); Accessor`
⇒ Return 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.

3. public boolean isEmptyCollection(); Observer

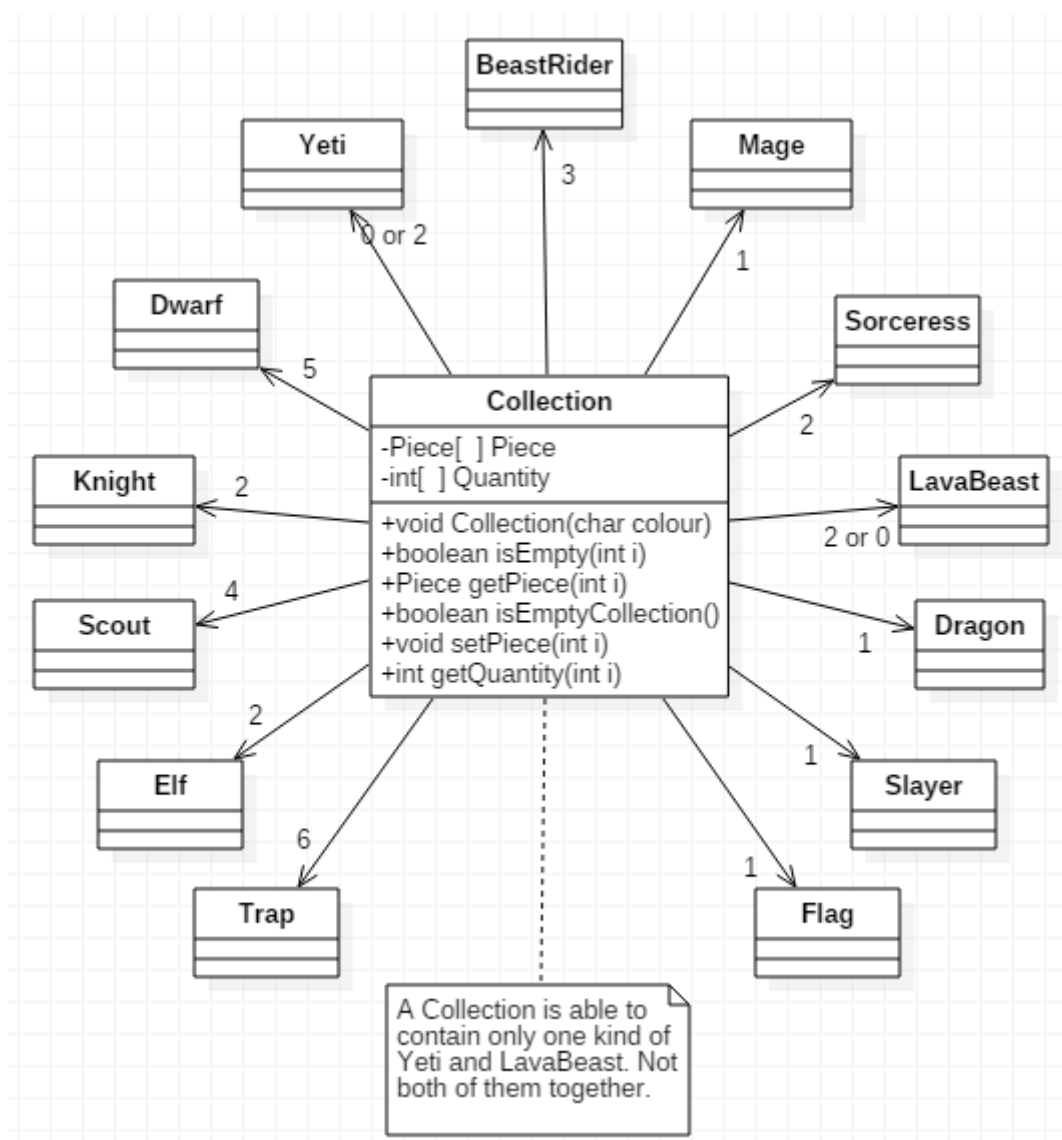
⇒ Return true if the collection is empty, or false different.

4. void setPiece(int i); Transformer

⇒ Rise the quantity(+1) of piece i

5. public int getQuantity(int i); Accessor

⇒ Return the quantity of piece i



Player

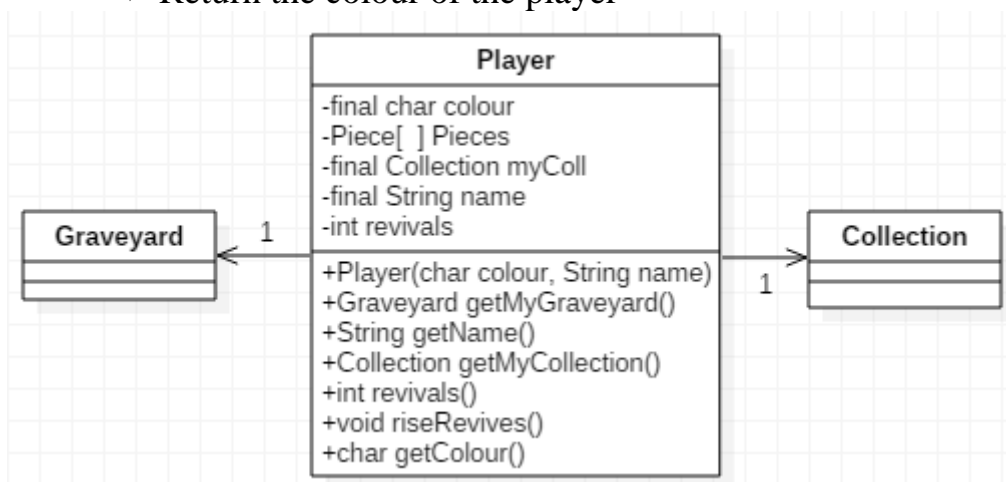
Η κλάση αυτή αναπαριστά κάθε παίκτη του παιχνιδιού.
Συνδέεται τόσο με ένα Collection όσο και με ένα Graveyard.

THE ATTRIBUTES:

1. final private char colour; // What colour player is
2. final private Collection myColl; // Standard collection with pieces
3. private Graveyard myGrave; // A graveyard for reviving of pieces
4. final private String name; // What its name
5. Private int revivals;

THE METHODS:

1. public int revivals(); Accessor
6. Return the summary of the revivals which has made
2. public void riseRevives();Transformer(mutative)
7. Rises the revivals(+1)
3. public String getName(); Accessor
⇒ Returns name of the player
4. public Collection getMyCollection(); Transformer(mutative)
⇒ returns a collection with all the pieces of the player.
5. public Graveyard getMyGraveyard(); Accessor
⇒ Returns a Graveyard with all the pieces of the player.
6. public char getColour(); Accessor
⇒ Return the colour of the player



Board

Η κλάση αυτή αναπαριστά το ταμπλό του παιχνιδιού.

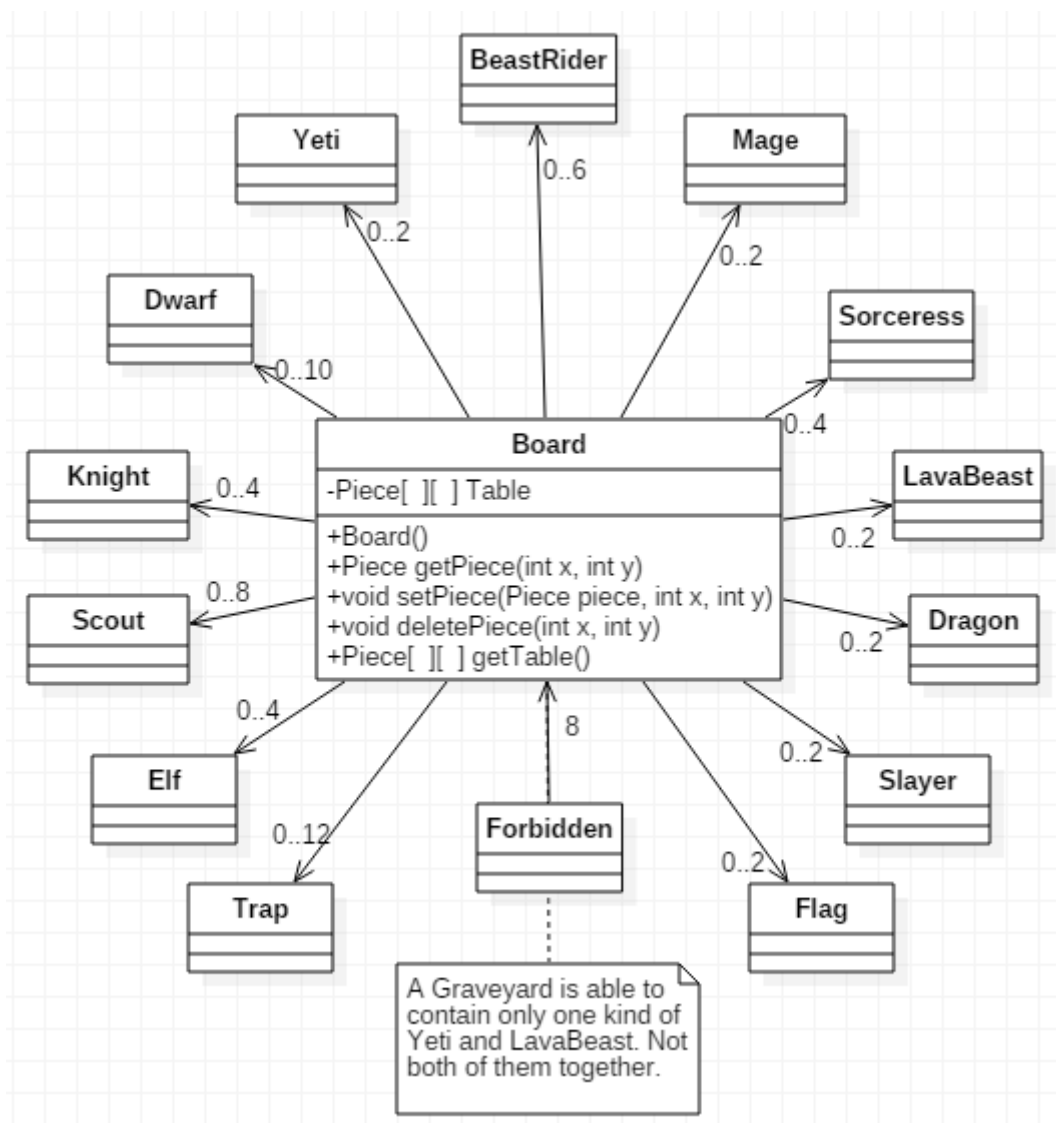
Επίσης, μας παρέχει μεθόδους για την τροποποίηση του.

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
3. public Piece[][] getTable(); Accessor
⇒ Return the table of the game
4. public void deletePiece(int x, int y); Transformer(mutative)
⇒ Delete a piece of the table



Music

Το πακέτο αυτό δεν περιλαμβάνει τίποτα ιδιαίτερο πέραν από μιας κλάσης η οποία περιέχει μια μέθοδο μέσω της οποίας παίζεται μουσική.

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.

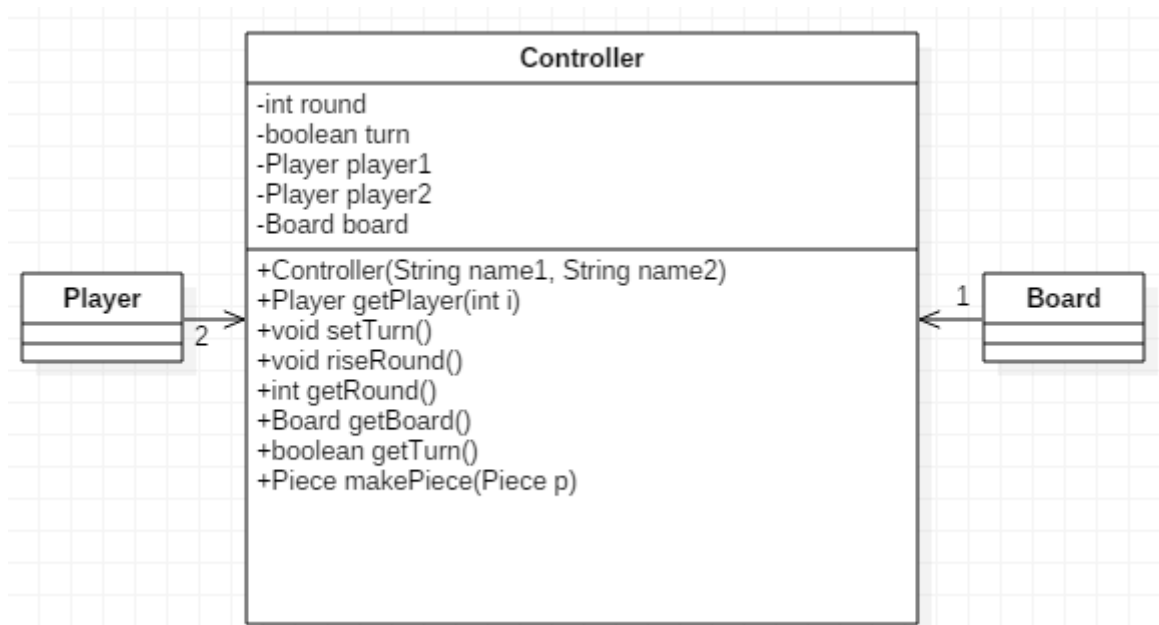
THE ATTRIBUTES:

1. private int round; // Hold the round which has passed
2. private int 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 Player getPlayer(int i); Accessor(selector)
⇒ Return a the player i
2. public void setTurn(); Transformer(mutative)
⇒ Change the turn of the game
3. public void riseRound(); Transformer(mutative)
⇒ Rise the number of the rounds of the game which have passed
4. public int getRound(); Accessor(selector)
⇒ Return the number of the rounds which have passed

5. `public Board getBoard(); Accessor(selector)`
⇒ Return the table of the game
6. `public boolean getTurn(); Observer`
⇒ Return who player's turn is
7. `public Piece makePiece(Piece p); Accessor(selector)`
⇒ Makes a clone of the piece and returns it



Package View

Το πακέτο αυτό περιλαμβάνει την κλάση **Graphics** που αναπαριστά το ταμπλό του παιχνιδιού ως γραφική απεικόνιση, το **graveyard** κάθε παίκτη καθώς επίσης και ένα μενού εντολών. Επίσης, εμπεριέχει κάποιες βοηθητικές μεθόδους για την ομαλή εκτέλεση του παιχνιδιού καθώς επίσης περιλαμβάνει και μερικούς **action listener** οι οποίοι υλοποιούν τις ενέργειες που επιθυμεί ο παίκτης να εκτελεστούν.

Graphics

THE ATTRIBUTES:

1. private JButton[][] Cell; // The table of the game
2. private JPanel NJPanel; // Includes all the elements of the game
3. private JButton setGame; // Both of players place their piece
4. private randomGame; // A random setting of the game
5. private mixGame; // Only one of the players places its pieces
6. private SGBHandler setGameHand;
 - ⇒ Action listener for preparing of a setting game
7. private RGBHandler randGameHand;
 - ⇒ Action listener for preparing of a random game
8. private MGBHandler mixGameHand;
 - ⇒ Action listener for preparing of a mix game
9. private ButtonHandler handler; // Action listener for moving of a piece
10. private Controller controller; // Link the graphics with the model
11. private Random in; // It's used for random pieces' placement
12. private int number; // Include the result of random number
13. private Piece piece; // Temp position of a piece
14. private Piece revP; // Temp position of a piece
15. private int moves; // Check if a player has chosen piece & place
16. private MixSorting mixSorting; // Action listener for placement of a piece
17. private SetSorting setSorting; // Action listener for placement of a piece
18. private JLabel jlabel; // Just a label
19. private Music mc; // Emulate the sound of some moves
20. private JButton[] BGrave // The graveyard of player Blue;
21. private JButton[] RGrave; // The graveyard of player Red;
22. private int posX, posY; // New coords of a piece which will be moved
23. private ReviveHandler reviveHandler;
 - ⇒ Action listener for moving of one revived piece
24. private Choice4ReviveHandler CFRHandler;
 - ⇒ Action listener for choosing defeated piece
25. private int prevTurn; //Hold the previous turn;

THE METHODS:

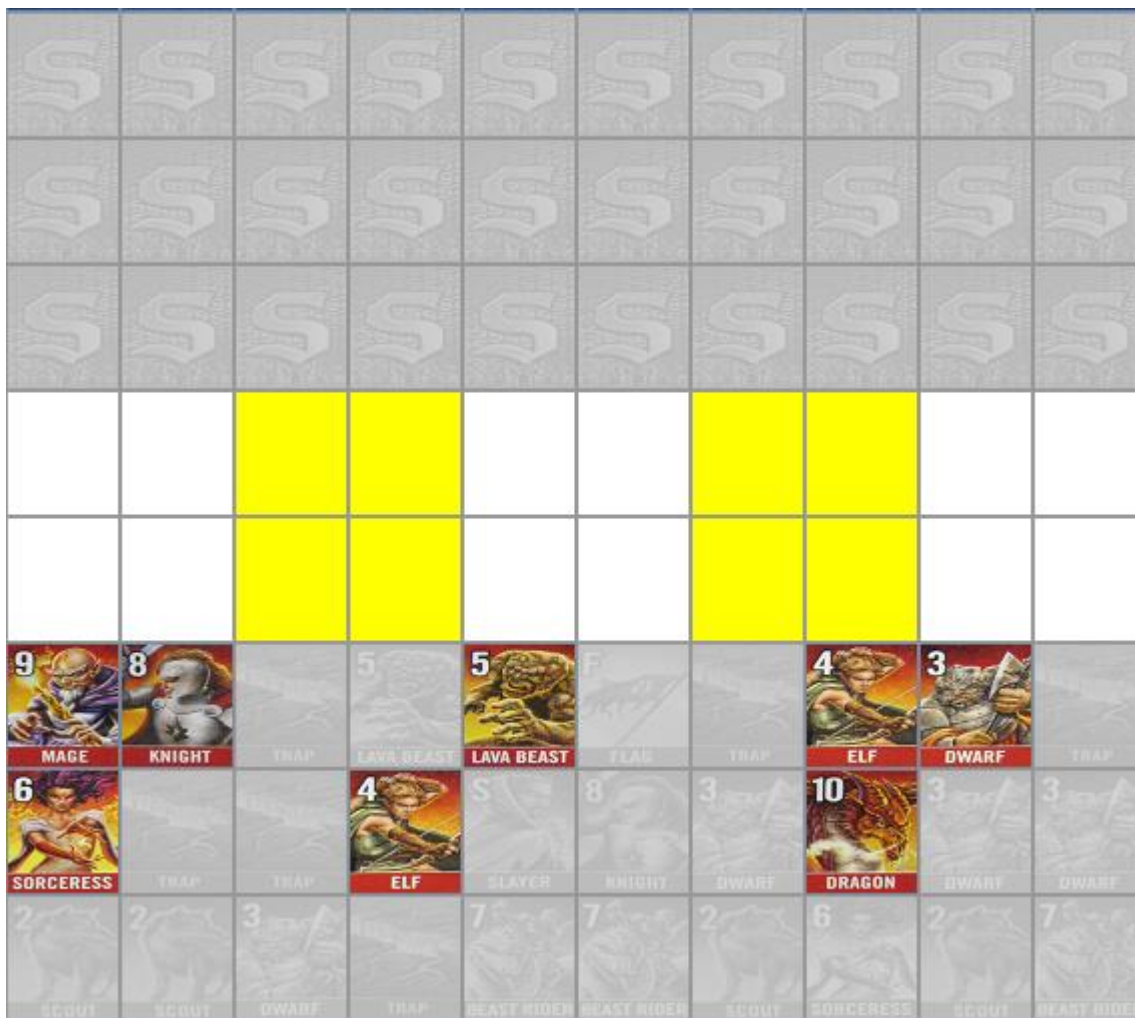
1. public void setHiddenPiece(); Transformer(mutative)
 - ⇒ Make the hidden pieces of the game
2. public void prepareTheTable(); Transformer(mutative)
 - ⇒ Prepare the game. Place the table and the graveyards into the panel
3. public void attack(int x, int y, int xx, int yy); Transformer(mutative)
 - ⇒ Prepare the game. Place the table and the graveyards into the panel

4. `public void makeChanges(); Transformer(mutative)`
⇒ Change the icon of the pieces, the statement and the background
5. `public void revive(int x, int y); Transformer(mutative)`
⇒ Places the piece which has revived
6. `public Piece makePiece(int i, char colour); Accessor(selector)`
⇒ Returns a new piece

Πιο συγκεκριμένα, 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.



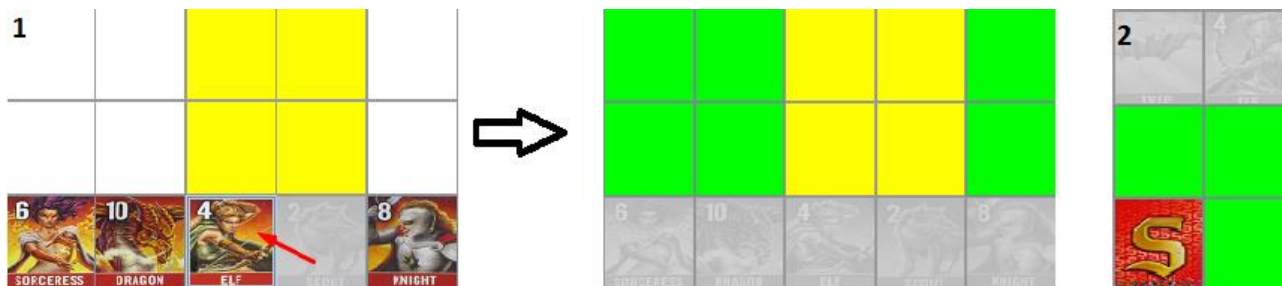
Ακόμη, θεμεριεύει ένα ταμπλό 8 X 10. Τα 30 αρχικά απ' αυτά ανήκουν στον έναν παίκτη και άλλα 30 στον άλλο. Περιέχει επίσης κάποιες ελεύθερες θέσεις κίνησης(12) και κάποιες απαγορευμένες(8). Επίσης περιλαμβάνει και κάποιες κρυμμένες(36), οι οποίες βοηθούν στην ομαλή εκτέλεση των κινήσεων. Οι θέσεις αυτές περικυκλώνουν τετραγωνικά το βασικό ταμπλό του παιχνιδιού. Στην γραφική απεικόνιση δεν εμφανίζονται οι κρυμμένες θέσεις παρά μόνο οι υπόλοιπες.



Όπως ευλόγως γίνεται αντιληπτό, από την πλευρά του παίκτη του οποίου είναι η σειρά, φαίνονται ενεργά μόνο τα πόνια τα οποία μπορούν να κινηθούν. Αντιθέτως, του αντιπάλου είναι τόσο ανενεργά όσο και γυρισμένα ανάποδα.

Αυτό διευκολύνει σε σύγκριση με το να ήταν όλα ενεργά αλλά δίχως action listener τόσο εμάς στην υλοποίηση όσο και τον χρήστη κατά την χρήση της εφαρμογής καθώς δεν χρειάζεται, αν δεν γνωρίζει (απόλυτα) τους κανόνες, να ελέγχει ένα ένα για να μάθει ποιο κινείται. Δυστυχώς όμως, δεν υποστηρίζεται λειτουργία αναίρεσης κίνησης. Άπαξ και πατηθεί κάποιο κουμπί, δεν μπορείς να το ακυρώσεις. Αυτό ισχύει σε οτιδήποτε σχετικό με αυτήν την υλοποίηση.

Όταν επιλεγεί ένα κουμπί, τότε όλα τα υπόλοιπα αδρανοποιούνται πλην όλων όσων μπορούν να είναι πιθανοί προορισμοί του κουμπιού-επιλογή μας. Τα κουμπιά αυτά γίνονται πράσινα αν είναι ελεύθερες θέσεις ή ενεργά αν εμπεριέχουν άλλα, αντίπαλα, πόνια.



Επιλέγεται το ξωτικό. Μπορεί να κάνει τετραγωνικά κίνηση με ακτίνα 2. Τα κίτρινα είναι απαγορευμένα, άρα δεν ενεργοποιούνται.

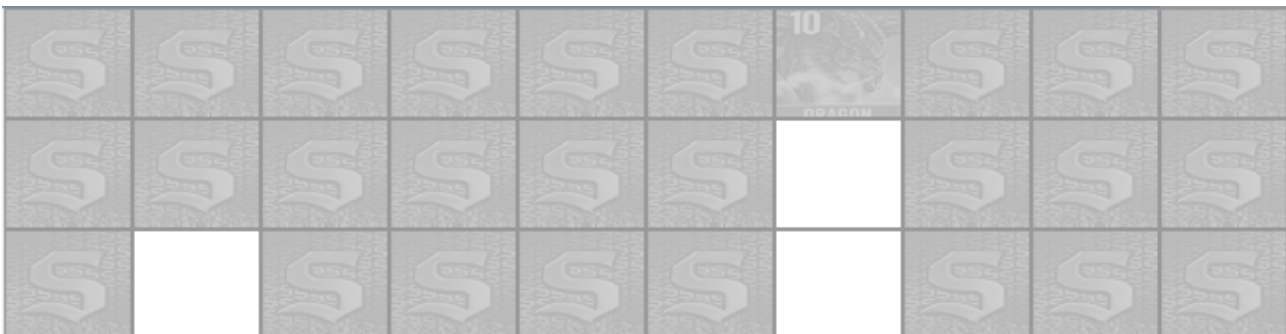
Τέλος υπάρχουν δύο υπό-ταμπλό των οποίων ο ρόλος είναι, τόσο να μας προβάλλουν ποια παιχνίδια έχουν βγει εκτός παιχνιδιού(ή εναλλακτικά, ποια διαθέτει ακόμη ο εκάστοτε παίκτης), όσο και να αξιοποιηθούν κατά την αναζωογόνηση κάποιου πιονιού, μέσω ενός άλλου πιονιού. Το λεγόμενο Graveyard. Τέλος, όλα τα κουμπιά, προστίθονται σε ένα panel.



Το graveyard του παίκτη με τα Blue Pieces. Εμφανίζονται ενεργά αλλά δεν υπακούν σε πατήματα όσο δεν είναι ώρα για κάποια τοποθέτηση.



Το graveyard του παίκτη με τα Red Pieces. Εμφανίζονται ενεργά αλλά δεν υπακούν σε πατήματα όσο δεν είναι ώρα για κάποια τοποθέτηση.



Το Dragon Piece έφτασε στην γραμμή τερματισμού. Δεν έχει κάνει ούτε το συγκεκριμένο κομμάτι επαναφορά ούτε ο κάτοχος του συνολικά.



Αυτό σημαίνει πως μπορεί να γίνει κάποια επαναφορά. Και όπως φαίνεται, υπάρχει ένα πiónι το οποίο βρίσκεται στο graveyard.

Ανάλογα με το σε πιο στάδιο βρίσκεται το παιχνίδι τοποθετούνται τα κατάλληλα, και στις κατάλληλες θέσεις, κουμπιά.

Τέλος, υπάρχει μια μπάρα, μέσω της οποίας δίνεται η δυνατότητα εκκίνησης ενός νέου παιχνιδιού. Το γραφικό παράθυρο επιστρέφει στην αρχική οθόνη με τις 3 επιλογές στησίματος των πιονιών.

SGB Handler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Ο ρόλος του είναι να ετοιμάσει το ταμπλό έτσι ώστε οι δύο παίκτες να τοποθετήσουν από μόνοι τους τα πιόνια τους. SGB = Set Game Button.

Set Sorting

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Ο ρόλος του είναι να λάβει τις επιλογές του παίκτη και να τοποθετήσει κατάλληλα τα πιόνια. Ακολουθείται αμέσως μετά την SGBHandler.

RGB Handler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Ο ρόλος του είναι να ετοιμάσει τυχαία το ταμπλό και το παιχνίδι να ξεκινήσει με βάση την προκαθορισμένη θέση των πιονιών. RGB = Random Game Button.

MGB Handler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Ο ρόλος του είναι να ετοιμάσει το ταμπλό έτσι ώστε ο ένας παίκτης να τοποθετήσει από μόνος του τα πιόνια του. Τα πιόνια του αντιπάλου είναι τυχαία καθορισμένα εκ των προτέρων. MGB = Mix Game Button.

Mix Sorting

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Ο ρόλος του είναι να λάβει τις επιλογές του παίκτη και να τοποθετήσει κατάλληλα τα πιόνια. Ακολουθείται αμέσως μετά την MGBHandler.

Choice4ReviveHandler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Χρησιμοποιείται μόνο όταν είναι ώρα για επαναφορά ενός πιονιού στο παιχνίδι. Ο ρόλος του είναι να λάβει την επιλογή πιονιού του παίκτη και να τοποθετήσει κατάλληλα το επιλεγμένο πιόνι.

ReviveHandler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Χρησιμοποιείται μόνο όταν είναι ώρα για επαναφορά ενός πιονιού στο παιχνίδι. Ο ρόλος του είναι να λάβει την επιλογή θέσης του παίκτη και να τοποθετήσει κατάλληλα το επιλεγμένο πιόνι. Ακολουθείται αμέσως μετά την Choice4ReviveHandler.

ButtonHandler

Η κλάση αυτή είναι εσωτερική της Graphics. Το μόνο που εμπεριέχει είναι έναν action listener. Χρησιμοποιείται μόνο όταν το παιχνίδι είναι έτοιμο προς εκκίνηση. Οποιαδήποτε μετακίνηση ή επίθεση πιονιού, ελέγχεται μέσω αυτού. Ακολουθείται από όλες τους προηγούμενους action listeners.

About Differences

- Μια σημαντική διαφορά σε σχέση με την πρώτη περιγραφική υλοποίηση είναι πως αφαιρέθηκε η εσωτερική κλάση MJButton, της κλάσης Graphics, η οποία έκανε extends την κλάση JButton. Αυτό οφειλόταν σε προβλήματα υλοποίησης του ίδιου του συστήματος. Θα μπορούσε να δημιουργηθεί μια κλάση που απλά περιέχει την άλλη αλλά παρ' όλ' αυτά, μιας και το σκέφτηκα αργά αυτό, ακολουθείται ο παραδοσιακός τρόπος «ψαξίματος» του ποιου κουμπιού πατήθηκε κάθε φορά.
- Επίσης, αρκετές μεθόδους είτε αντικαταστήθηκαν, είτε αλλάξανε όνομα για λόγους κομψότητας είτε προστέθηκαν. Η μεγαλύτερη και κύρια αλλαγή, όπως είναι λογικό, έγινε στην κλάση Graphics μέσα στην οποία προστέθηκαν επίσης

πάρα πολλά attributes όπως και εσωτερικές κλάσεις – action listeners.

- Ο βασικός κορμός της σχεδίασης - υλοποίησης παρέμεινε ο ίδιος. Δεν έγινε καμία απολύτως αφαίρεση παρά μόνο μιάς επιπλέον προσθήκη, αυτή της κλάσης Music όπως και του πακέτου sounds έτσι ώστε να γίνει το παιχνίδι περισσότερο ρεαλιστικό.
-