

**Αντικειμενοστρεφής
Προγραμματισμός HY252
Project 2015 – 2016**



Όνομα: Τίτος Σταύρος Χανιωτάκης

A.M.: 3351

Σχεδιασμός του Project

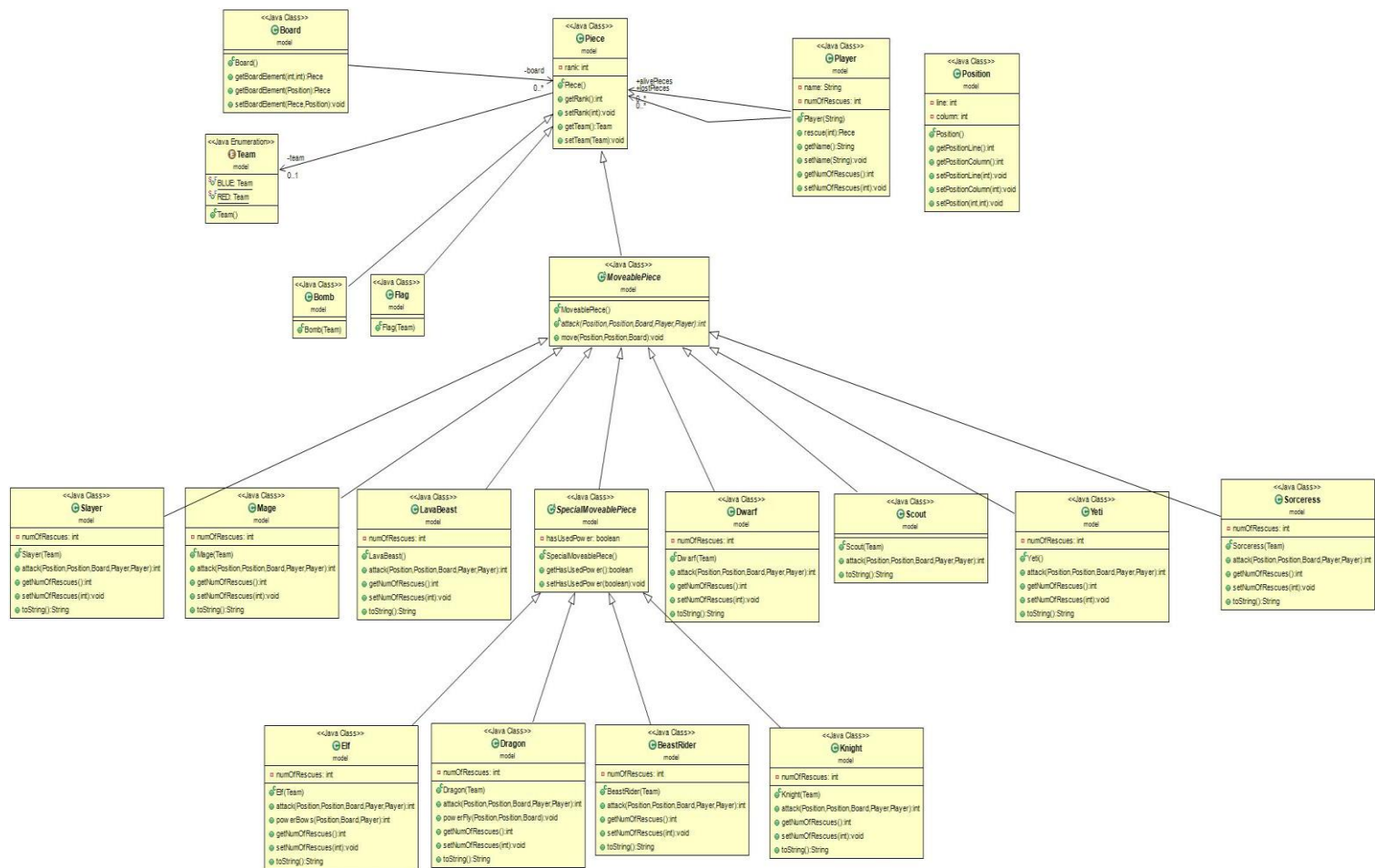
Βασική αρχή στην ανάπτυξη της παρούσας εργασίας είναι το μοντέλο MVC. Σύμφωνα με αυτό το πρότυπο διαχωρίζεται η ανάπτυξη της γραφικής διεπαφής (View) από τον πυρήνα του παιχνιδιού που περιέχει όλη την πληροφορία κατάστασης (Model) και από τον μηχανισμό διαχείρισης και ενημέρωσης των ενεργειών του παιχνιδιού με τη γραφική του απεικόνιση (Controller). Συνεπώς ο Controller είναι ο συνδετικός κρίκος ανάμεσα στο Model και το View.

Package Model

Σε αυτό το πακέτο περιέχονται:

- η κλάση Piece
- οι υποκλάσεις της Piece (η abstract κλάση MoveablePiece, η κλάση Bomb και η κλάση Flag)
- οι υποκλάσεις της MoveablePiece (η abstract κλάση SpecialMoveablePiece, η κλάση Slayer, η κλάση Mage, η κλάση LavaBeast, η κλάση Dwarf, η κλάση Scout, η κλάση Yeti, η κλάση Sorceress)
- οι υποκλάσεις της SpecialMoveablePiece (η κλάση Elf, η κλάση Dragon, η κλάση BeastRider, η κλάση Knight)
- η κλάση Board
- η Enumeration κλάση Team
- η κλάση Player
- η κλάση Position

Εδώ θα δείξουμε μια αναπαράσταση των κλάσεων του πακέτου Model μέσω UML.



Η κλάση Piece

Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private int rank; //The rank of the Piece
2. private Team team; //The team that Piece belongs

Οι μέθοδοι:

1. `public int getRank() //Accessor (Selector)`
Returns the Piece's rank.
2. `public void setRank(int rank) //Transformer(Mutative)`
Sets the Piece's rank.
3. `public Team getTeam() // Accessor (Selector)`
Returns the Piece's team.
4. `public void setTeam(Team team) //Transformer (Mutative)`
Sets the Piece's team.

Αφαίρεσα τη boolean μεταβλητή `defeated` συνεπώς και τις μεθόδους `setDefeated` και `getDefeated` καθώς κατέληξα στο γεγονός ότι δεν θα μου χρειαστεί.

Κλάσεις που κληρονομούν την `Piece`:

- Οι κλάσεις `Bomb` και `Flag`
- Η κλάση `MoveablePiece`

Οι κλάσεις `Bomb` και `Flag`

Οι κλάσεις `Bomb` και `Flag` κληρονομούν τα attributes και τις μεθόδους που έχει η κλάση `Piece` και δεν έχουν επιπλέον attributes ή μεθόδους.

Η κλάση MoveablePiece

Η κλάση MoveablePiece κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση Piece και έχει κάποια επιπλέον attributes και κάποιες επιπλέον μεθόδους.

Οι μέθοδοι:

1. `public abstract void attack(Position position1, Position position2, Board board, Player player1, Player player2)`
`//Transformer`
The piece attacks another piece.
2. `public void move(Position position1, Position position2, Board board)` `//Transformer`
The piece moves.
3. `public ArrayList<Position> possibleMovePositions()`
`//Transformer`
Returns an ArrayList with the possible positions the piece can move.

Αφαίρεσα τις boolean μεταβλητές `hasAttacked`, `hasMoved` συνεπώς και τις μεθόδους `getHasAttacked`, `setHasAttacked`, `getHasMoved` και `setHasMoved` καθώς κατέληξα στο γεγονός ότι δεν θα μου χρειαστούν.

Κλάσεις που κληρονομούν την MoveablePiece:

- Οι κλάσεις `Slayer`, `Mage`, `LavaBeast`, `Dwarf`, `Scout`, `Yeti` και `Sorceress`.
- Η κλάση `SpecialMoveablePiece`

Οι κλάσεις Slayer, Mage, LavaBeast, Dwarf, Yeti, Sorceress.

Οι κλάσεις Slayer, Mage, LavaBeast, Dwarf, Yeti, Sorceress κληρονομούν τα attributes και τις μεθόδους που έχει η κλάση MoveablePiece και έχουν κάποια επιπλέον attributes και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. private int numOfRescues; //The number of rescues this Piece has done.

Οι μέθοδοι:

1. public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer
The piece attacks another piece.
2. public int getNumOfRescues //Accessor (Selector)
Returns the value of the variable numOfRescues.
3. public void setNumOfRescues(int numOfRescues)
//Transformer (Mutative)
Sets the value of the variable numOfRescues.

Αφαίρεσα τη μέθοδο rescue καθώς ήταν πιο εύκολο να την έχω μέσα στον Controller.

Η κλάση Scout

Η κλάση Scout κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση MoveablePiece και έχει μία επιπλέον μέθοδο.

Οι μεθόδοι:

1. `public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer`
The piece attacks another piece.

Η κλάση SpecialMoveablePiece

Η κλάση SpecialMoveablePiece κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση MoveablePiece και έχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. `private boolean hasUsedPower; //If the piece has used it's power.`

Οι μεθόδοι:

1. `public boolean getHasUsedPower() //Observer`
Returns the value of the variable hasUsedPower.
2. `public void setHasUsedPower(boolean hasUsedPower) //Transformer(Mutative)`
Sets the value of the variable hasUsedPower.

Κλάσεις που κληρονομούν την SpecialMoveablePiece:

- Οι κλάσεις Elf, Dragon, BeastRider και Knight.

Η κλάση Elf

Η κλάση Elf κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση SpecialMoveablePiece και έχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. private int numOfRescues; //The number of rescues this Piece has done.

Οι μέθοδοι:

1. public void rescue(Piece piece, ArrayList<Piece> lostPieces)
//Transformer
This piece rescues a piece.
2. public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer
The piece attacks another piece.
3. public int getNumOfRescues //Accessor (Selector)
Returns the value of the variable numOfRescues.
4. public void setNumOfRescues(int numOfRescues)
//Transformer (Mutative)
Sets the value of the variable numOfRescues.

Αφαίρεσα τη μέθοδο powerBows καθώς ήταν πιο εύκολο να την έχω μέσα στον Controller.

Η κλάση Dragon

Η κλάση Dragon κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση SpecialMoveablePiece και έχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. private int numOfRescues; //The number of rescues this Piece has done.

Οι μέθοδοι:

1. public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer
The piece attacks another piece.
2. public int getNumOfRescues //Accessor (Selector)
Returns the value of the variable numOfRescues.
3. public void setNumOfRescues(int numOfRescues)
//Transformer (Mutative)
Sets the value of the variable numOfRescues.

Αφαίρεσα τις μεθόδους rescue και powerFly καθώς ήταν πιο εύκολο να τις έχω μέσα στον Controller.

Η κλάση BeastRider

Η κλάση BeastRider κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση SpecialMoveablePiece και έχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. private int numOfRescues; //The number of rescues this Piece has done.

Οι μεθόδοι:

1. public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer
The piece attacks another piece.
2. public int getNumOfRescues //Accessor (Selector)
Returns the value of the variable numOfRescues.
3. public void setNumOfRescues(int numOfRescues)
//Transformer (Mutative)
Sets the value of the variable numOfRescues.

Αφαίρεσα τις μεθόδους rescue και powerSpeed καθώς ήταν πιο εύκολο να τις έχω μέσα στον Controller.

Η κλάση Knight

Η κλάση Knight κληρονομεί τα attributes και τις μεθόδους που έχει η κλάση SpecialMoveablePiece και έχει ένα επιπλέον attribute και κάποιες επιπλέον μεθόδους.

Τα attributes:

1. private int numOfRescues; //The number of rescues this Piece has done.

Οι μεθόδους:

1. `public void attack(Position position1, Position position2, Board board, Player player1, Player player2) //Transformer`
The piece attacks another piece.
2. `public int getNumOfRescues //Accessor (Selector)`
Returns the value of the variable `numOfRescues`.
3. `public void setNumOfRescues(int numOfRescues) //Transformer (Mutative)`
Sets the value of the variable `numOfRescues`.

Αφαίρεσα τις μεθόδους `rescue` και `powerSpeed` καθώς ήταν πιο εύκολο να τις έχω μέσα στον `Controller`.

Η κλάση Board

Η κλάση `Board` αναπαριστά τον πίνακα του παιχνιδιού. Θα είναι ένας 8x10 πίνακας και μέσα σε αυτόν θα τοποθετούνται τα `Pieces`. Εδώ θα αναφέρουμε τα `attributes` και τις μεθόδους που έχει η κλάση αυτή.

Τα `attributes`:

1. `private Piece board[][]; //The board of the game is an array 8x10`

Οι μεθόδους:

1. `public Piece getBoardElement(int i, int j) //Accessor (Selector)`
Returns an element of the board.
2. `public Piece getBoardElement(Position position) //Accessor (Selector)`

Returns an element of the board.

3. `public void setBoardElement(Piece piece, Position position)`
//Transformer (Mutative)
Sets an element of the board.

Πρόσθεσα τη μέθοδο `getBoardElement` με ορίσματα δύο ακεραίους γιατί σε πολλές περιπτώσεις δεν είχα κάποια μεταβλητή τύπου `Position` αλλά δύο ακεραίους.

Η κλάση Player

Η κλάση `Player` αναπαριστά τον παίκτη. Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. `private String name;` //The name of the Player.
2. `private int numOfRescues;` //The number of rescues this Piece has done.
3. `public Piece[] alivePieces;` //An array with the pieces that are still alive.
4. `public ArrayList<Piece> lostPieces;` //An ArrayList with the pieces that have died.

Οι μέθοδοι:

1. `public void finishedTurn()` // Transformer
Make some settings when player finish his turn.
2. `public String getName()` // Accessor (Selector)
Returns the name of the Player.
3. `public void setName(String name)` // Transformer (Mutative)
Sets the name of the Player.

4. `public int getNumOfRescues()`
Returns the value of the variable `numOfRescues`.
5. `public void setNumOfRescues(int numOfRescues)`
Sets the value of the variable `numOfRescues`.

Αφαίρεσα τη `boolean` μεταβλητή `hasPlayed`, συνεπώς και τις μεθόδους `getHasPlayed` και `setHasPlayed`, καθώς τελικά δεν μου χρειάστηκε. Επίσης αφαίρεσα την `int` μεταβλητή `numOfLostPieces`, συνεπώς και τις μεθόδους `getNumOfLostPieces` και `setNumOfLostPieces`, καθώς τελικά δεν μου χρειάστηκε. Ακόμα αφαίρεσα την μέθοδο `finishedTurn` καθώς το τέλος της σειράς το διαχειρίστηκα μέσα στην `GraphicUI`.

Η κλάση Position

Η κλάση `Position` αναπαριστά την θέση στον πίνακα καθώς περιέχει δύο `int`, ένα για τη γραμμή και ένα για τη στήλη. Εδώ θα αναφέρουμε τα `attributes` και τις μεθόδους που έχει η κλάση αυτή.

Τα `attributes`:

1. `private int line; //The line of the position.`
2. `private int column; //The column of the position.`

Οι μέθοδοι:

1. `public int getPositionLine() //Accessor (Selector)`
Returns the line of the position.
2. `public int getPositionColumn() // Accessor (Selector)`

Returns the column of the position.

3. `public void setPositionLine(int line) // Transformer (Mutative)`

Sets the line of the position.

4. `public void setPositionColumn(int column) // Transformer (Mutative)`

Sets the column of the position.

5. `public void setPosition(int line, int column) // Transformer (Mutative)`

Sets the position.

Η κλάση Team

Είναι μία enum κλάση που περιέχει τις ομάδες {BLUE, RED}.

Package Controller

Σε αυτό το πακέτο περιέχεται μόνο η κλάση Controller.

Η κλάση Controller

Η κλάση Controller είναι ουσιαστικά αυτή που οργανώνει όλο το παιχνίδι. Είναι υπεύθνη για τη δημιουργία ενός νέου παιχνιδιού, τη δημιουργία στιγμιotypών παικτών, των πιονιών του κάθε παίκτη και τη σύνδεση μεταξύ του View και του Model. Αυτό που κάνει η συγκεκριμένη κλάση είναι να δέχεται τις επιλογές του χρήστη από το View και να πραγματοποιεί οποιαδήποτε ενέργεια χρειάζεται ώστε να παίζεται σωστά το

παιχνίδι. Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private Board board; //The board of the game.
2. private Player p1,p2; //The two players.
3. private int Round; //The round of the game.
4. Position board_position; //A Position variable
5. int random_i, random_j; //Are used for find random numbers.

Οι μέθοδοι:

1. public Piece choice(Position pos) //Accessor (Selector)
Returns a player's choice from the board.
2. public void PlayTurn(ExtendedButton button1, ExtendedButton button2)
//Transformer
Plays the turn of the player.
3. public void Rescue(Position pos1, Piece piece) // Transformer
The piece with position pos1 rescue a Piece.

Αφαίρεσα τη μέθοδο choice καθώς τελικά δεν μου χρειάστηκε. Επίσης αφαίρεσα τις μεθόδους PlayMove, PlayAttack, PlayFly, PlaySpeed και PlayBows και πρόσθεσα μία μέθοδο PlayTurn όπου ανάλογα με τα δύο κλικ του χρήστη η μέθοδος αυτή ξεχωρίζει αν θα κάνει κίνηση, επίθεση ή θα χρησιμοποιήσει κάποια ειδική δύναμη. Επίσης αφαίρεσα την μέθοδο TurnFinished καθώς διαχειρίστηκα το τέλος της σειράς στην GraphicUI.

Εδώ θα δείξουμε μία αναπαράσταση της κλάσης Controller μέσω UML.



Package View

Σε αυτό το πακέτο περιέχονται οι κλάσεις `ExtendedButton`, `GraphicUI`, `NewExitDialog`, `RescueDialog`.

Η κλάση `ExtendedButton`

Η κλάση `ExtendedButton` εξειδικεύει την κλάση `JButton`. Περιέχει ακόμα δύο μεταβλητές τύπου `int` όπου αποθηκεύουν τις συντεταγμένες του κάθε κουμπιού μέσα στο `GridLayout`.

Η κλάση GraphicUI

Η κλάση GraphicUI εξειδικεύει την κλάση JFrame. Δημιουργεί ένα panel και μέσα σε αυτό ένα GridLayout (8, 10) το οποίο είναι το ταμπλό του παιχνιδιού και αποτελείται από 80 κουμπιά τύπου ExtendedButton. Όταν πατιέται ένα κουμπί αν δεν είναι άδριο το συγκεκριμένο κελί του πίνακα αποθηκεύεται η επιλογή του χρήστη. Όταν πατιέται το δεύτερο κουμπί καλείται από τον Controller η μέθοδος PlayTurn που αποφασίζει αν είναι έγκυρη η επιλογή και αποφασίζει τι κίνηση θα κάνει σύμφωνα με τις 2 επιλογές του χρήστη. Στην 1η φάση σκεφτόμουν να φτιάξω κουμπιά Move, Attack, Fly, Speed, Attack with Bows, Rescue αλλά δεν το υλοποίησα με αυτό τον τρόπο γιατί θα ήταν πιο πολύπλοκο για το χρήστη.

Η κλάση RescueDialog

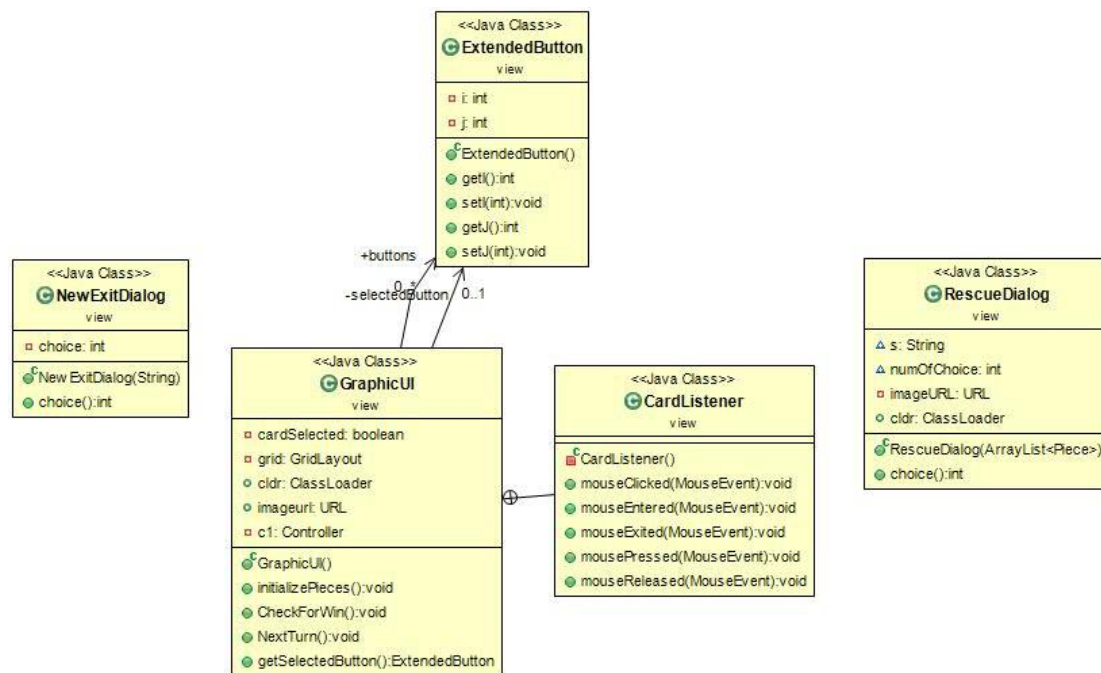
Στο ίδιο πακέτο υπάρχει και μία κλάση RescueDialog η οποία δημιουργεί ένα frame. Στο frame αυτό υπάρχει ένα JOptionPane το οποίο εμφανίζεται όταν κάποιο πiónι κάνει μία διάσωση, έτσι θα εμφανίζονται τα πiónια που δεν είναι πια στο παιχνίδι έτσι ώστε ο παίκτης να μπορεί να επιλέξει ποιο επιθυμεί να διασώσει. Στην κλάση αυτή υπάρχει μία μέθοδος choice() που επιστρέφει το πiónι που ο παίκτης επιθυμεί να διασώσει.

Η κλάση NewExitDialog

Η κλάση αυτή δημιουργεί ένα frame και μέσα στο frame υπάρχει ένα JOptionPane το οποίο εμφανίζεται σε περίπτωση

νίκης ενός από τους δύο παίκτες. Έτσι εμφανίζεται το μήνυμα “Player1/Player2 won! Do you want to play a new game?” και ο χρήστης αποφασίζει αν θα πατήσει Yes ή No για να ξεκινήσει ένα καινούριο παιχνίδι. Στην κλάση αυτή υπάρχει μία μέθοδος choice() που επιστρέφει την επιλογή του χρήστη.

Εδώ θα δείξουμε μία αναπαράσταση του πακέτου View μέσω UML.



Τέλος στο πακέτο View υπάρχει μία ακόμα κλάση, η Main στην οποία υπάρχει μία main όπου δημιουργείται ένα στιγμιότυπο της GraphicUI και έτσι ξεκινάει το παιχνίδι.