



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2022-23

**Stratego®**

*Νικολέττα Αρβανίτη*

**4844**

# Εισαγωγή

## Περιεχόμενα

<i>1. Εισαγωγή.....</i>	<i>2</i>
<i>2. Η σχεδίαση και οι κλάσεις του πακέτου Model.....</i>	<i>2-11</i>
<i>3. Η σχεδίαση και οι κλάσεις του πακέτου Controller.....</i>	<i>11-12</i>
<i>4. Η σχεδίαση και οι κλάσεις του πακέτου View.....</i>	<i>13-14</i>
<i>5. Διάγραμμα UML.....</i>	<i>15</i>

## 1. Εισαγωγή

Η υλοποίηση της εργασίας είναι βασισμένη πάνω στο μοντέλο MVC (Model View Controller). Άρα σκοπός είναι οι πληροφορίες να διαχωρίζονται σε τρία μέρη. Ο Controller ουσιαστικά συνδέει το Model και το View, όπου στην συνέχεια της αναφοράς θα αναλύσουμε περαιτέρω την λειτουργία τους.

## 2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Σε αυτό πακέτο περιέχονται: η Abstract κλάση Piece, οι κλάσεις ImmovablePiece και MovablePiece οι οποίες κληρονομούν την Piece, η κλάση Board, η κλάση Player και οι κλάσεις Dragon, Mage, Dwarf, Sorceress, Knight, LavaBeast, Scout, Slayer, Yeti και Elf που επεκτείνουν την MovablePiece αλλά και οι Bomb, Flag που επεκτείνουν την ImmovablePiece.

### Piece Abstract Class and other classes for Piece

#### Τα Attributes:

1. Private String name; // the name of a Piece
2. Private int value; // the value of a Piece
3. Private char colour; // the colour of the piece. R for red and B for blue
4. Private Icon icon; // the icon of the Piece

#### Αυτή η abstract κλάση παρέχει τις εξής μεθόδους:

1. Public Piece(char colour); Constructor  
Constructs a new Piece with the given colour
2. Public Piece(); Constructor  
Constructs a simple Piece

3. Public char getColour( ); Accessor (Selector)

Returns the colour of the piece

4. Public void setIcon(String icon); Transformer (Mutative)

Sets the piece's icon

5. Public void setColour(char colour); Transformer (Mutative)

Sets the colour of the piece

6. Public Icon getIcon( ); Accessor (Selector)

Returns the icon of the piece

7. Public int getValue( ); Accessor (Selector)

Returns the value of the piece

8. Public void setValue(int value); Transformer (Mutative)

Sets the value of the piece

9. Public String getName( ); Accessor (Selector)

Returns the name of the piece

10. Public void setName(String name); Transformer (Mutative)

Sets the name of the piece

Στην συνέχεια έχουμε την MovablePiece και ImmonablePiece που κληρονομούν την Piece.

### **Class MovablePiece**

Εδώ θα αναφερθούν τα attributes και οι υπόλοιπες μέθοδοι που χρησιμοποιεί αυτή η κλάση (εκτός από αυτές που υλοποιεί μέσω της κλάσης Piece)

#### **Τα Attributes:**

- 1.Private int rescues; // the number of rescues of the piece
- 2.Private int value; // the value of the piece
- 3.Private String name; // the name of the piece

### **Αυτη η κλάση παρέχει τις εξής μεθόδους:**

1. Public MovablePiece(char colour) Constructor

Constructs a new movable piece with given colour and zero rescues

Έπειτα κάνει override μεθόδους της κλάσης Piece.

Οι υπόλοιπες κλάσεις που κληρονομούν την MovablePiece απλά περιέχουν έναν constructor που δέχεται το χρώμα και με την εντολή super το βάζουν στην μεταβλητή.

### **Class ImmoveablePiece**

Αυτή η κλάση δεν έχει επιπλέον attributes και μεθόδους, απλά διαθέτει δυο constructors, έναν για τα Immoveable Pieces με χρώμα και έναν για τα Forbidden pieces.

### **Class Player**

#### **Τα Attributes:**

1. Private String name; // the name of the player
2. Private int captures; // the number of the player's captures
3. Private int rescues; // the number of the player's rescues

4. Private int successfulAttacks; // the successful attacks of the player
5. Private int totalAttacks; // the total number of the player's attacks
6. Private boolean isAlive; // true if player alive, false otherwise
7. Public Piece[ ] Pieces; // an array with all the pieces of the player
8. Final Private char colour; // the colour of the player
9. Final private Collection collection; // the collection of the pieces

**Αυτή η κλάση παρέχει τις εξής μεθόδους:**

1. Public Player(char colour, String name);  
Constructor  
  
Constructs a new player with a name and sets all the attributes
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 void setCaptures(int Captures);  
Transformer (Mutative)

Sets the captures of the player

5. Public int getCaptures( ); Accessor (Selector)

Returns the captures of the player

6. Public int getRescues( ); Accessor (Selector)

Returns the rescues of the player

7. Public void setRescues(int Rescues);  
Transformer (Mutative)

Sets the rescues of the player

8. Public int getSuccessfulAttacks( ); Accessor  
(Selector)

Returns the successful attacks of the player

9. Public void setSuccessfulAttacks(int  
successfulAttacks); Transformer (Mutative)

Sets the successful attacks of the player

10. Public void setTotalAttacks(int totalAttacks);  
Transformer (Mutative)

Sets the total attacks of the player

11. Public int getTotalAttacks( ); Accessor  
(Selector)

Returns the total attacks of the player

12. Public int percentOfSuccess( );

Returns the player's success percent

13. Public void addAttack( ); Transformer  
(Mutative)

Adds an attack to the total attacks

14. Public void addSuccessfulAttack( );  
Transformer (Mutative)

Adds a successful attack to the successful attacks

15. Public void setIsAlive(boolean isAlive);  
Transformer (Mutative)

Sets the isAlive to true if player is alive, false otherwise

16. Public boolean getIsAlive( ); Accessor (Selector)

Returns the boolean isAlive, true if player is alive and false otherwise

17. Public Piece[ ] getPieces( ); Accessor (Selector)

Returns the pieces of the player

18. Public char getColour( ); Accessor (Selector)

Returns the colour of the player

19. Public Collection getCollection( ); Accessor (Selector)

Returns the collection of the player

Στην συνέχεια έχουμε την κλάση Board που δημιουργεί τις συνθήκες του board στο οποίο θα κινούμαστε.



## **Class Board**

### **Ta Attributes**

1. Piece [ ][ ] board; // the 8x10 board we need

### **Αυτή η κλάση παρέχει τις εξής μεθόδους:**

1. Public Board( ); Constructor

Constructs a new board 8x10 for the game

2. Public void killHero(int x, int y);

Removes the piece from the board

3. Public Piece [ ][ ] getBoard( ); Accessor (Selector)

Returns the board of the game

4. Public void setPiece(Piece p, int x, int y);

Sets a piece at the given x and y coordinates

5. Public boolean isMovable(int x1, int y1, int x2, int y2);

Checks if we can move the piece of the x1 and y1 to x2 and y2

6. Public boolean isInstanceOfFlag(Piece p);

Checks if piece is instance of flag

7. Public boolean isInstanceOfTile(Piece p);

Checks if piece is instance of tile

8. Public boolean isInstanceOfBomb(Piece p);

Checks if piece is instance of bomb

9. Public boolean isInstanceOfDwarf(Piece p);

Checks if piece is instance of Dwarf

10. Public boolean isInstanceOfImmovable(Piece p);

Checks if piece is instance of ImmovablePiece

11. Public boolean isInstanceOfDragon(Piece p);

Checks if piece is instance of Dragon

12. Public boolean isInstanceOfSlayer(Piece p);

Checks if piece is instance of slayer

13. Public void setTile(int x, int y);

Sets a new tile at the given coordinates

14. Public Piece getPiece(int x, int y);

Returns the piece at the given coordinates

## **Class Collection**

### **Τα Attributes**

1. Private Piece[ ] Piece; // array of the collection's pieces
2. Private int[ ] Quantity; // array of the quantity of the pieces

### **Αυτή η κλάση παρέχει τις εξής μεθόδους:**

1. Public Collection(char colour); Constructor  
Constructs a new collection of pieces
2. Public boolean isEmpty(int I); Observer  
Returns true if there isn't quantity, false otherwise
3. Public boolean isEmptyCollection( ); Observer  
Returns false if there is at least a piece
4. Public int getQuantity(int I); Accesor(Selector)  
Returns the quantity of piece in position
5. Public Piece getPiece(int I); Accessor (Selector)  
Returns the piece in position I

6. Public void setPiece(int I;  
Transformer(Mutative)

Rise the quantity of piece I

7. Public int returnPos(Piece p);

Returns the position of the given piece inside  
the array

### 3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

#### Controller

Η κλάση Controller κάνει την σημαντικότερη δουλειά, καθώς όπως λέει και το όνομα της, διαχειρίζεται τις επιλογές του παίκτη μέσα από την γραφική απεικόνιση και εκτελεί τις κατάλληλες λειτουργίες μέσα στο model. Παρακάτω θα αναφερθούν τα attributes καθώς και οι μέθοδοι που χρησιμοποιεί.

#### Ta Attributes:

1. Private Player fire, ice; // the two player of the game
2. Private boolean GameOver; // false if game is still on  
and true if game is over
3. Private Board board; // a board for the game(8x10)
4. Private int PlayerTurn=0; // the player turn

## **Οι μέθοδοι:**

1. Controller( String name1, String name2); Constructor

Constructs a new game with two players and a board

2. Public void setGameOver( ); Transformer (Mutative)

Sets the variable GameOver to true

3. Public boolean getGameOver( ); Accessor (Selector)

Returns true if game is over and false otherwise

4. Public void rescue( Piece p, int x, int y);

This method brings a piece back from the dead and puts it on the board at (x,y)

5. Public void setPlayerTurn( ); Transformer

Sets the turn to the next player

6. Public int getPlayerTurn( ); Accessor

Returns which player's turn is right now

7. Public Player getPlayer( ); Accessor

Returns the player that is playing right now

8. Public Board getBoard( ); Accessor

Returns the board of the game

## 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

### Package view

Αυτό το πακέτο περιλαμβάνει την κλάση Graphs και MenuDialog. Η κλάση Graph θα δημιουργεί ένα Board 8x10, θα καλύπτει και τις απαγορευμένες ζώνες του, αλλά και θα βάζει τα πιόνια στις θέσεις που χρειάζεται. Θα περιλαμβάνει την γραφική αναπαράσταση όλου του παιχνιδιού. Η κλάση MenuDialog θα δημιουργεί το μενού στην αρχή του παιχνιδιού όπου ο χρήστης θα διαλέγει αν επιθυμεί μειωμένο στρατό ή/και καμία υποχώρηση.

### Class Graphs

#### Οι μέθοδοι:

##### 1. Public Graphs(); Constructor

Constructs a new board of JButtons, sets everything in place, including icons, mouse listeners etc.

##### 2. Private ImageIcon getInageCard(int num, String team);

Returns the icon that should be in the JButton when it's the player's turn

3. Private ImageIcon getHiddenImageCard(int playerTurn);

Returns the hidden image of the other player's cards

4. Public void mouseClicked(MouseEvent e);

Plays the most important part of the class, selects the buttons, moves the pieces, checks for special cases etc.

5. Public int getXCoordinate(JButton b);

Returns x coordinate of the JButton on the board

6. Public int getYCoordinate(JButton b);

Returns y coordinate of the JButton on the board

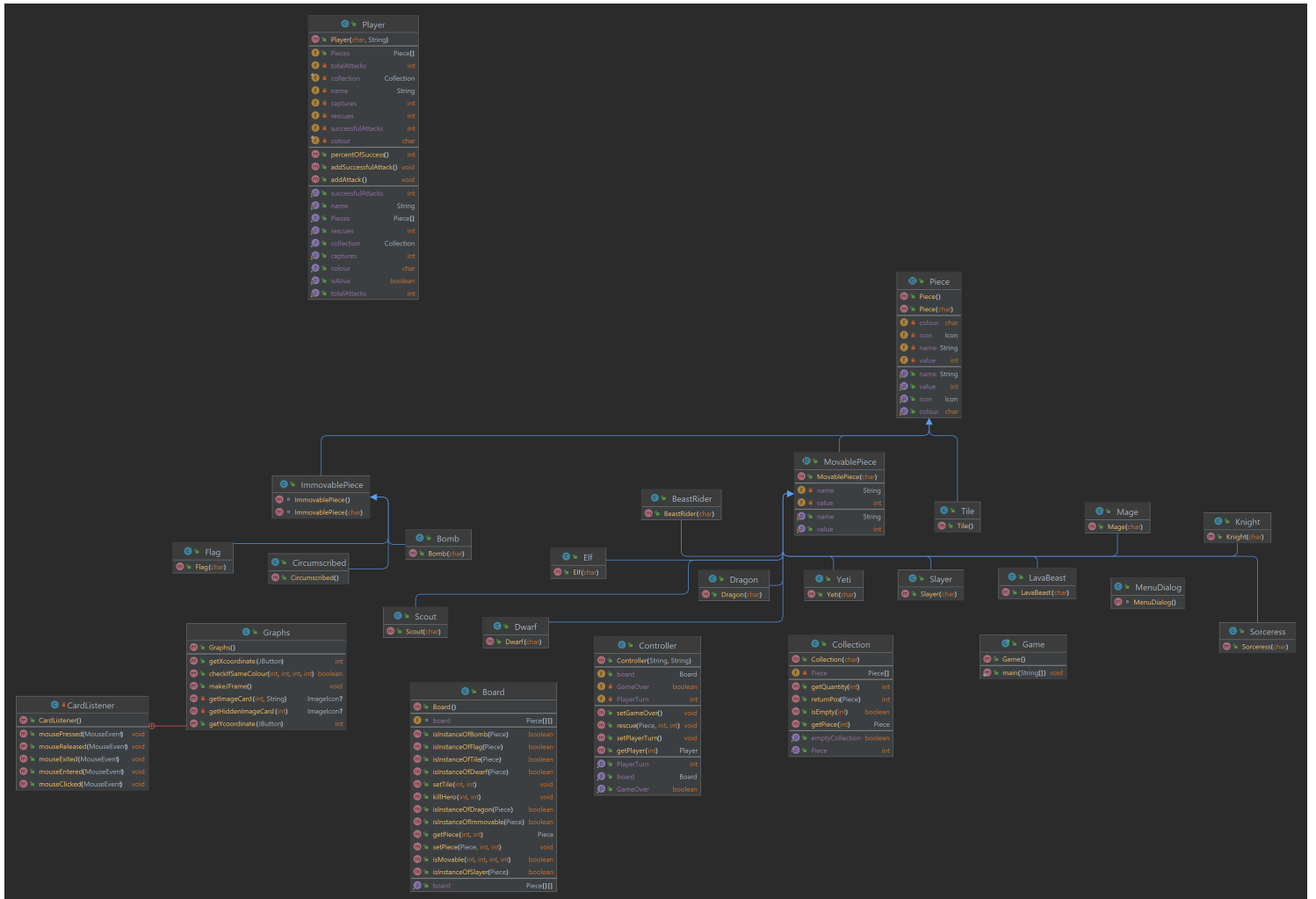
7. Public void MakeJFrame();

Makes a JFrame when the game is over and terminates the program

8. Public boolean checkIfSameColour(int x1, int y1, int x2, int y2);

Checks if the pieces that are at the given coordinates have the same colour

## 5. Διάγραμμα UML





## 6. Λειτουργικότητα

Με τα δεδομένα που έχει ο κώδικας αυτή τη στιγμή μπορεί να υποστηρίξει ένα κανονικό παιχνίδι Stratego, τοποθετώντας τα πιόνια κάθε φορά σε τυχαίες θέσεις, με σωστές εναλλαγές παικτών, ορθές κινήσεις όλων των πιονιών και νίκη όταν κατακτηθεί η σημαία. Δυστυχώς δεν υλοποιήθηκε το μενού και τα δυο διαφορετικά game modes που χρειαζόντουσαν.

## 7. Συμπεράσματα

Μέσα από την εργασία έμαθα πάρα πολλά για την συγκεκριμένη γλώσσα προγραμματισμού. Ήταν η πρώτη φορά που είδα πως υλοποιούνται τα γραφικά ενός παιχνιδιού και έμαθα να εκτιμώ λίγο παραπάνω τα παιχνίδια με γραφικό περιβάλλον διότι απαιτούν μεγάλη προσπάθεια και θέληση για να πραγματοποιηθούν.