



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2021-2022

*Csd4111*

*10-12-22*

## Περιεχόμενα

1.	Εισαγωγή .....	1
2.	Η Σχεδίαση και οι Κλάσεις του Πακέτου Model.....	1
3.	Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller .....	5
4.	Η Σχεδίαση και οι Κλάσεις του Πακέτου View .....	6
5.	Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML .....	7
6.	Λειτουργικότητα (Β Φάση).....	8
7.	Συμπεράσματα.....	9

## Εισαγωγή

*Το πρόγραμμα που θα υλοποιήσουμε θα γίνει σύμφωνα με το μοντέλο MVC.Όπου η υλοποίηση θα είναι ένα επιτραπέζιο παιχνίδι το Stratego.Σκοπός μας είναι να το δημιουργήσουμε ενωνοντας το Controler με το Model και το View.*

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

Σε αυτό το πακετο θα αναλύσουμε το πακετο του Montel τις κλασεις που θα παρεχει interfaces κτλ.

### **Class Player**

Όπου θα αναλύσουμε αρχικά τα γνωρίσματα της κλασης:

- 1.private Color color; /player color
- 2.private String name/player name
- 3.private boolean Turn;/player turn

Με μεθοδους:

*-public Color getColor() /accessor return the color of Pieces that has the Player*

-*public void* setColor(Color color)/transformer set the color of the Pieces that has the Player

-*public String* getName()/accesor return the Name of the Player

-*public void* setName(String name) / transformer set the name of the player

-*public boolean* isTurn()/accesor which player has turn to play

-*public void* setTurn(boolean turn)/ transformer set the turn of the player

### **Κλασεις και interfaces που χρησιμοποιουντε για τα Pieces**

Εχουμε αρχικα το abstract class Piece οπου διαχωριζεται σε MoveablePiece (τα Piece που μπορούν να κινηθούν) καθώς και τα ImmoveablePiece (που δεν κινούνται). Επίσης έχω υλοποίηση και μια κλάση SpecialMoveablePiece που κληρονομούν κλάσεις με Pieces που έχουν κάποιο special move.

### **Abstract Class Piece**

Με γνωρίσματα:

1.private Color color /το χρωμα του piece

2.private String img / η εικονα για το piece

3.private int num/αριθμος καθε καρτας αναλογα τη δυναμη της

Και μεθόδους:

-*public Color* getColor() /accesor *return the color of the Piece*

-*public void* setColor(Color color) / transformer set the color of the Piece

Την οποια την κληρονομουν οι εξης κλασεις:

1.MoveablePiece

2.ImmoveablePiece

### **Class ImmoveablePiece**

Με μεθόδους:

-

Την οποία την κληρονομούν οι εξής κλάσεις:

- 1.Bomb
- 2.Flag

### **Class MovablePiece**

Με γνωρίσματα κλάσης:

1. private int num; /οπου num είναι ο αριθμός που έχουν τα movable piece

Και μεθόδους:

*-public int getNum() /accessor return the Number of each movable Piece*

*-public void setNum(int num) / transformer set for each movable Piece the Number*

Την οποία την κληρονομούν οι εξής κλάσεις:

- 1.SpecialMoveablePiece
- 2.Sorceress
- 3.Mage
- 4.LavaBeast
- 5.Knight
- 6.Elf
- 7.Dragon
- 8.BeastRider

### **Class SpecialMoveablePiece**

Με μεθόδους:

-

Την οποία την κληρονομούν οι εξής κλάσεις:

1.Slayer

2.Scout

3.Dwarf

### **Κλάσεις που υλοποιούνται για το Board**

#### **Class Board**

Με γνωρίσματα κλάσης:

- 1.Piece[][] pieces;//οπου ειναι το καθε piece
- 3.private Piece[] shuffle;//shuffle για τα μπλε
- 4.private Piece[] shuffle1;//shuffle για τα κοκκινα

Και μεθόδους:

- public Piece[][] getArray()/ *accesor return the array of Pieces*
- public void Initialize\_Board()/ *initialize the board for the blue and red images*
  
- public Piece[][] newLocation(int fromx,int fromy,int x,int y)/ *it change the location of the card to the new x,y*
- public Piece[][] newLocation1(int fromx,int fromy,int x,int y)/ *newLocation1 is when you attack i check if the card is beaten or not and if the card attack to the flag you win*
- public boolean newLocationResult()/ *@return true or false if you win or not to set what color should the card have to set after the back image*

Κλάσεις που την κληρώνουν:

-

### **Κλάσεις που υλοποιούνται για το Player**

#### **Class Player**

Με γνωρίσματα κλάσης:

- 1.private Color color;/το χρώμα θα είναι ένα από τα χαρακτηριστικά του player
- 2.private String name;/ονόμα player
- 3.private boolean Turn;/η σειρά του player

Και μεθόδους:

*-public Color getColor() /accessor the color of Pieces that has the Player*

*-public void setColor(Color color)/ transformer set the color of the Pieces that has the Player*

*-public String getName() /accessor return the Name of the Player*

*-public void setName(String name)/ transformer set the name of the player*

*-public boolean isTurn() /accessor return which player has turn to play*

*-public void setTurn(boolean turn)/ transformer set the turn of the player*

Κλάσεις που την κληρονομούν:

-

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

### **Class Controller**

Η κλάση αυτή είναι το κύριο μέρος του προγράμματος αφού από αυτή ελεγχεται .Επίσης είναι αυτή που συνδέει τα δυο άλλα κομμάτια model&view.

Με γνωρίσματα κλάσης:

- 1.private CreatePlayer player;;
- 2.private Board array;

Και με μεθόδους:

- public CreatePlayer getPlayer() / *accessor return if player1 starts*
- public Board getArray()/ *transformer that initialize the board game*
- public void start\_Turn()/ *start\_Turn ,take random who is playig first red or blue*

//τις υπολοιπες παρακατω δεν τις εχω υλοποιησει

-void initialize\_Pawns() / *transformer that initialize the Pawns*

-void initialize\_Pieces()/ *transformer that initialize the Pieces*

-void initialize\_Board( Piece[][] array)/ *transformer that initialize the board game*

-*boolean* is\_attacker\_winning(Piece attacker,Piece defender)/ *observe return if the attacker wins*

-*boolean* is\_move\_alloed(Board board,int xFrom,int yFrom,int xTo,int yTo)/ *observe return if the Piece of the possition that is now if he can move or not*

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

#### **Class View**

Εχουμε ότι το view θα είναι το πακετο που θα υλοποιει τα γραφικα .Αρα θα εχουμε κλασεις που υλοποιουν τα γραφικα. Το κυριο πανελ ειναι το View οπου θα είναι το JFrame μας .Επισης θα εχουμε 4 πανελ(Κλασεις) ένα για την αρχικοποιηση του Board,και αλλα 3 panel (Κλασεις)οπου θα εχει τα Info για τα (Ενεργοι κανονες ,Στατιστικα,Αιχμαλωτισεις) και μεσα σε αυτά τα αντιστοιχα texts και edit texts για το κάθε πεδιο.Επισης στο Board panel θαεχουμε τις καρτες που θα είναι υλοποιημενα ως Buttons.

Στη κλαση View θα συνδεσουμε πανω τα υπολοιπα πανελς

Αλλα θα προσθεσουμε και τη συναρτηση refresh οπου κανει refresh το board

#### **Class Board panel**

Η κλαση η οποια υλοποιηται το board του παιχνιδιου ,αλλα και που περιεχει πολλα περισσοτερα.Αρχικα γινεται μεσω της συναρτησεις createboard η αρχικοποιηση του αρχικου board για τις αναλογες θεσεις που πρεπει να μπει το κάθε image, επισης

εχουμε τις getImageBackBlue getImageBackRed οπου παρνει τις εικονες για το background.

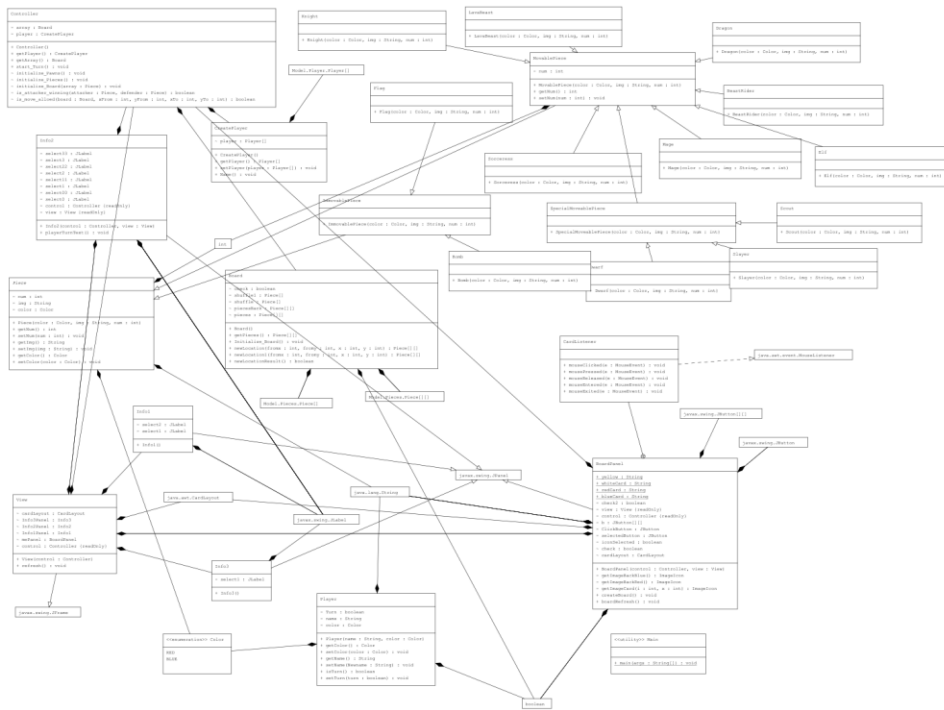
Επίσης υλοποιείται και το actionlistener όπου αν επιλεγεί καρτά και στη συνέχεια η καινούργια θέση γίνεται η αλλαγή της θέσης, καθώς υλοποιείται και για το που θα πρέπει να μετακινηθεί η καρτά αναλογα με τα green border που έχουν τοποθετηθεί τα red border που εμφανίζονται είναι για τις καρτες που δεν μπορούν να μετακινηθούν.

Η συνάρτηση boardRefresh είναι αυτή που διαβαζει το board με τις καινουργιες θεσεις πλεον αφου καλουμε τις newLocation συναρτησεις και γινεται η αλλαγη της καινουργια θεσεις.

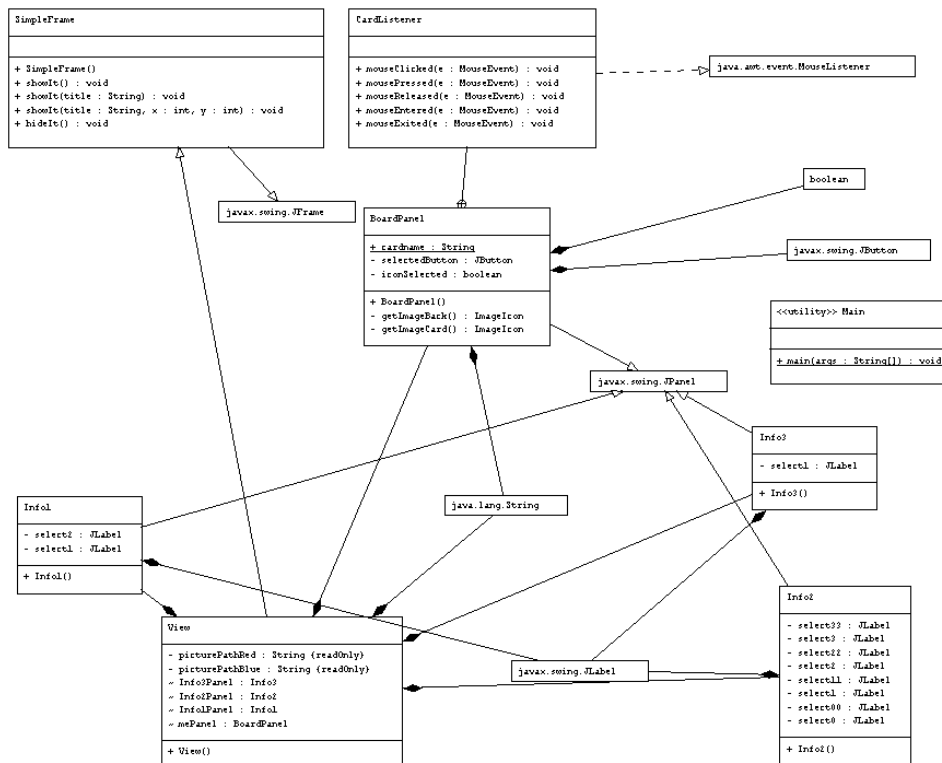
#### 4. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

## UML Β φάση





## UML\_VIEW α φάση



## 5. Λειτουργικότητα (Β Φάση)

Όσο αναφορά τη λειτουργικότητα έχω γράψει παραπάνω για το πώς υλοποιείται το παιχνίδι.