



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2019-2020

[Project Java - Β φάση]

[Φανουράκης Νικόλαος]

[4237]

[15/01/2020]

Περιεχόμενα

<u>1. Εισαγωγή</u>	3
<u>2. Αλλαγές σε σχέση με την Α φάση</u>	3
<u>3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model</u>	4
<u>4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View</u>	16
<u>5. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller</u>	18
<u>6. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML</u>	20
<u>7. Λειτουργικότητα</u>	23

- Εισαγωγή

Σε αυτήν την αναφορά θα δούμε τον σχεδιασμό και την υλοποίηση του παιχνιδιού Sorry!

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller). Οι κλάσεις του Model θα περιέχουν τα δεδομένα που μας είναι απαραίτητα για να υλοποιήσουμε το παιχνίδι. Η κλάση View θα αποτελεί την γραφική διεπαφή του παιχνιδιού. Τέλος, η κλάση Controller θα αποτελεί τον συνδετικό κρίκο μεταξύ View και Model και θα ενημερώνει το παιχνίδι σε κάθε αλλαγή. Ακολουθούν περισσότερες πληροφορίες για τις κλάσεις και τις λειτουργίες τους.

- Αλλαγές σε σχέση με την Α φάση

Προσθήκες:

1. Συναρτήσεις changeTurn και newGame στη κλάση Controller.
2. Συναρτήσεις startAgain και reachHome στη κλάση View.
3. Συνάρτηση sendStart στη κλάση Card.
4. Πεδίο JLabel tableBackground και συναρτήσεις setTableBackground και getInfoBox στη κλάση View.
5. Συναρτήσεις ableToMove και movePawn με 3 παραμέτρους στη κλάση NumberSevenCard.

Αλλαγές:

1. Μεταφορά συνάρτησης fold από την κλάση Deck στην κλάση Controller.
2. Μεταφορά συνάρτησης showWinningMessage από την κλάση View στην κλάση Controller.
3. Οι PawnListener, CardListener, MenuListener, FoldListener θα υλοποιούν το interface MouseListener αντί για ActionListener.
4. Η συνάρτηση updatePawn της κλάσης View θα παίρνει 4 ορίσματα (τα 4 buttons πιόνια) και θα ενημερώνει τη θέση όλων των πιονιών στο παιχνίδι.

- Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

- Πακέτο Model.Card

Σε αυτό το πακέτο θα περιέχονται οι abstract κλάσεις Card και NumberCard η οποία κληρονομεί την Card , η κλάση SorryCard η οποία επίσης κληρονομεί την Card και οι κλάσεις SimpleNumberCard, NumberOneCard, NumberTwoCard, NumberFourCard, NumberSevenCard, NumberTenCard, NumberElevenCard οι οποίες κληρονομούν την κλάση NumberCard.

Abstract Κλάση Card

Η δημιουργία μιας abstract κλάσης Card η οποία θα κληρονομείται, μας δίνει τη δυνατότητα να ορίσουμε τα κοινά χαρακτηριστικά όλων των καρτών και να έχουμε πρόσβαση στα δεδομένα τους χωρίς να χρειάζεται να ξέρουμε τι είδους κάρτα έχει παιχτεί.

Η κλάση αυτή έχει τα εξής πεδία :

1)private boolean isPlayed; // Let us know if the card is played.

2)private String image; //The path of the image of the card.

Και περιέχει τις εξής μεθόδους :

1)public String getImage(); //Returns a string describing the path of the image.(Accessor)

2)public void setImage(String image); //Sets the path of the image of the card.(Transformer)

3)public void DropCard(); //Sets card as played.(Transformer)

4)public void putBackCard(); //Sets card as unplayed.(Transformer)

5)public boolean IsPlayed(); //Informs if the card is already played.(Observer)

6)public abstract String toString(); //Informs about the attributes of the card.(Accessor)

7)public abstract void movePawn(Pawn pawn,Deck table);

8)public abstract boolean ableToMove(Pawn pawn,Deck table);

9)public void sendStart(Pawn pawn,Deck table); //Sets pawn to its start position.(Transformer)

Οι abstract συναρτήσεις θα υλοποιηθούν από τις υποκλάσεις της Card.

Στη συνέχεια έχουμε τις κλάσεις NumberCard και SorryCard που κληρονομούν την Card.

Abstract κλάση NumberCard

Τα attributes:

1)private int number; //The number of the card

Οι μέθοδοι:

1)public void setNumber(int number); //Sets the number of the card.(Transformer)

2)public int getNumber(); //Gets the number of the card.(Accessor)

3)public void movePawn(Pawn pawn,Deck table); //Moves the pawn as much as the number of the card.(Transformer)

4)public boolean ableToMove(Pawn pawn,Deck table); //Checks if the pawn can be moved front as much as the number of the card.(Observer)

Ακολουθούν οι υποκλάσεις της NumberCard.

Κλάση SimpleNumberCard

Η κλάση αυτή αναπαριστά τις κάρτες 3,5,8,12 οι οποίες έχουν απλό τρόπο για να κινούν το πιόνι.

Με την κάρτα αυτή το πιόνι κινείται μπροστά τόσες θέσεις όσες και ο αριθμός της, εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση.

Μεθόδους:

1)public SimpleNumberCard(int number);//Creates an object of type SimpleNumberCard and sets it with the given number.(Constructor)

2)public String toString(); //Informs about the attributes of the card.

Κλάση NumberOneCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 1.

Με την κάρτα αυτή το πιόνι μπορεί να βγεί από το Start Square ή να μετακινηθεί 1 τετράγωνο μπροστά ,εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση.

Μεθόδοι:

1)public NumberOneCard(int number);//Creates an object of type NumberOneCard and sets it with number 1.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawn(Pawn pawn, Deck table); //Changes the position of the pawn by 1 or moves it out of its start square.(Transformer)

4)public boolean ableToMove(Pawn pawn, Deck table); //Checks if the pawn can be moved out of its square or 1 square front.(Observer)

Κλάση NumberTwoCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 2.

Με την κάρτα αυτή το πιόνι μπορεί να μετακινηθεί έξω από το Start Square ή 2 θέσεις μπροστά, εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση. Επίσης ο παίχτης που έχει το πιόνι ξαναπαίζει.

Μεθόδοι:

1)public NumberTwoCard(int number);//Creates an object of type NumberTwoCard and sets it with number 2.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawn(Pawn pawn, Deck table); //Changes the position of the pawn by 2 or moves it out of its start square.(Transformer)

4)public boolean ableToMove(Pawn pawn, Deck table); //Checks if the pawn can be moved out of its square or 2 squares front.(Observer)

Κλάση NumberFourCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 4.

Με αυτή τη κάρτα το πιόνι μετακινείται 4 θέσεις πίσω, εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση.

Μεθόδοι:

1)public NumberFourCard(int number); //Creates an object of type NumberFourCard and sets it with number 4.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawn(Pawn pawn, Deck table); //Moves the pawn 4 squares back.(Transformer)

4)public boolean ableToMove(Pawn pawn, Deck table); //Checks if the pawn can be moved 4 squares back.(Observer)

Κλάση NumberSevenCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 7.

Με την κάρτα αυτή ο παίχτης κινεί τα πιόνια του κατά 7 θέσεις σύνολο,εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτές τις θέσεις.

Μεθόδοι:

1)public NumberSevenCard(int number); //Creates an object of type NumberSevenCard and sets it with number 7.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawn(Pawn pawn1, int increasePosition1, Pawn pawn2, int increasePosition2, Deck table); //Moves player's pawns 7 squares front in total.(Transformer)

4)public boolean ableToMove(Pawn pawn1, int increasePosition1, Pawn pawn2, int increasePosition2, Deck table); //Checks if this move is valid.(Observer)

5)public void movePawn(Pawn pawn, int increasePosition, Deck table); //Moves pawn by increasePosition.(Transformer)

6)public boolean ableToMove(Pawn pawn, int increasePosition, Deck table);
//Checks if pawn can move by increasePosition.(Observer)

Κλάση NumberTenCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 10.

Με την κάρτα αυτή το πιόνι μετακινείται 10 θέσεις μπροστά ή 1 θέση πίσω, εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση.

Μεθόδοι:

1)public NumberTenCard(int number); //Creates an object of type NumberTenCard and sets it with number 10.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawnBack(Pawn pawn, Deck table); //Moves the pawn 1 square back.(Transformer)

4)public boolean ableToMoveBack(Pawn pawn, Deck table); //Checks if this move is valid.(Observer)

Κλάση NumberElevenCard

Η κάρτα αυτή αναπαριστά την κάρτα με το νούμερο 11.

Με την κάρτα αυτή ο παίχτης μετακινεί ένα πιόνι του 11 θέσεις μπροστά εάν δεν υπάρχει άλλο πιόνι ίδιου χρώματος σε αυτή τη θέση. Αλλιώς αν θέλει μπορεί να ανταλλάξει θέσεις μεταξύ ενός πιονιού του και ενός πιονιού αντιπάλου με την προϋπόθεση ότι τα πιόνια δεν βρίσκονται στις θέσεις start, home ή safety zone.

Μεθόδοι:

1)public NumberElevenCard(int number); //Creates an object of type NumberElevenCard and sets it with number 11.(Constructor)

2)public String toString(); //Informs about the attributes of the card.(Accessor)

3)public void movePawn(Pawn pawn, Pawn opponentPawn, Deck table); //Exchanges positions between player's pawn and opponent's pawn.(Transformer)


```
4)public boolean ableToMove(Pawn pawn,Pawn opponentPawn, Deck table);  
//Checks if this move is valid.(Observer)
```

Ακολουθεί η άλλη υποκλάση της Card η SorryCard.

Κλάση SorryCard

Η κάρτα αυτή αναπαριστά την κάρτα Sorry.

Με την κάρτα αυτή ο παίχτης ανταλλάσει θέσεις μεταξύ ενός πιονιού του και ενός πιονιού του αντιπάλου με την προϋπόθεση ότι τα πιόνια δεν βρίσκονται στις θέσεις start, home και safety zone.

Μεθόδοι:

```
1)public String toString(); //Informs about the attributes of the card.(Accessor)
```

```
2)public void movePawn(Pawn pawn,Pawn opponentPawn, Deck table); //Exchanges  
positions between player's pawn and opponent's pawn.(Transformer)
```

```
3)public boolean ableToMove(Pawn pawn,Pawn opponentPawn, Deck table);  
//Checks if this move is valid.(Observer)
```

```
4)public void movePawn(Pawn pawn,Deck table);
```

```
5)public boolean ableToMove(Pawn pawn,Deck table);
```

Στην συνέχεια έχουμε την abstract κλάση Square και τις υποκλάσεις της.

- Πακέτο Model.Square

Abstract Κλάση Square

Η δημιουργία μιας abstract κλάσης Square η οποία θα κληρονομείται, μας δίνει τη δυνατότητα να ορίσουμε τα κοινά χαρακτηριστικά όλων των τετραγώνων του ταμπλό και να έχουμε πρόσβαση στα δεδομένα τους χωρίς να χρειάζεται να ξέρουμε σε ποιο τετράγωνο βρισκόμαστε.

Πεδία:

- 1)private Pawn pawnOn; //The pawn which is on the square.
- 2)private int position; //The position of the square.
- 3)private Color color;//The color of the square.

Μεθόδοι:

- 1)public void setPosition(int position); //Sets the position of the square.(Transformer)
- 2)public void setColor(Color color); //Sets the color of the square.(Transformer)
- 3)public void setPawnOn(Pawn pawn); //Sets a pawn on the square.(Transformer)
- 4)public int getPosition(); //Returns the position of the square.(Accessor)
- 5)public Color getColor(); //Returns the color of the square..(Accessor)
- 6)public Pawn getPawnOn(); //Returns the pawn which is on the square..(Accessor)
- 7)public boolean isPawnOn(); //Informs about the existence of a pawn on a square.(Observer)

Κλάση HomeSquare

Η τελική θέση των πιονιών στο ταμπλό.

Μεθόδοι:

- 1)public HomeSquare(int position, Color color); //Creates an object of type HomeSquare and sets its position and color.(Constructor)

Κλάση StartSquare

Η αρχική θέση των πιονιών στο ταμπλό.

Μεθόδοι:

1)public StartSquare(int position, Color color); //Creates an object of type StartSquare and sets its position and color.(Constructor)

Abstract Κλάση SlideSquare

Την χρησιμοποιώ για την οργάνωση των υποκλάσεων της. Ακολουθούν αυτές οι υποκλάσεις.

Κλάση StartSlideSquare

Τα αρχικά τετράγωνα από τα σύνολα των τετραγώνων Slide.

Μεθόδοι:

1)public StartSlideSquare(int position, Color color); //Creates an object of type StartSlideSquare and sets its position and color.(Constructor)

Κλάση InternalSlideSquare

Τα ενδιάμεσα τετράγωνα από τα σύνολα των τετραγώνων Slide.

Μεθόδοι:

1)public InternalSlideSquare(int position, Color color); //Creates an object of type InternalSlideSquare and sets its position and color.(Constructor)

Κλάση EndSlideSquare

Τα τελευταία τετράγωνα από τα σύνολα των τετραγώνων Slide.

Μεθόδοι:

1)public EndSlideSquare(int position, Color color); //Creates an object of type EndSlideSquare and sets its position and color.(Constructor)

Κλάση SafetyZoneSquare

Τα τετράγωνα ασφαλείας.

Μεθόδοι:

1)public SafetyZoneSquare(int position, Color color); //Creates an object of type SafetyZoneSquare and sets its position and color.(Constructor)

Κλάση SimpleSquare

Τα υπόλοιπα απλά τετράγωνα του ταμπλό.

Μεθόδοι:

1)public SimpleSquare(int position, Color color); //Creates an object of type SimpleSquare and sets its position and color.(Constructor)

- Οι υπόλοιπες κλάσεις του πακέτου Model.

Κλάση Pawn

Κλάση για τα πιόνια των παιχτών.

Πεδία:

1)private Color color; //the color of the pawn.

2)private int position; //the position of the pawn.

3)private boolean enabled; //When it is true it means pawn is located at its start or safety zone square.

4)private boolean finished; //When it is true it means pawn reached its home square.

Μεθόδοι:

1)public Pawn(Color color,int position);//Creates an object of type Pawn and sets its position and color.(Constructor)

2)public boolean isEnabled(); //Informs us about the state of the field enabled.(Observer)

3)public void setDisable(); //Changes field enabled to false.(Transformer)

4)public void setEnable(); //Changes field enabled to true.(Transformer)

5)public void setPosition(); //Sets the position of the pawn.(Transformer)

6)public int getPosition(); //Returns the position of the pawn.(Accessor)

7)public void setFinished(); //Changes field finished to true;(Transformer)

8)public boolean isFinished(); //Informs about the state of the field finished.(Observer)

9)public Color getColor(); //Returns the color of the pawn. (Accessor)

Κλάση Player

Κλάση για του παίχτες.

Πεδία:

- 1)private String name; //The name of the player.
- 2)private Color color; //The color of the player/of the player's pawns.
- 3)private Pawn pawn1; //The first pawn of the player.
- 4)private Pawn pawn2; //The second pawn of the player.
- 5)private boolean turn; //When it is true it is player's turn to play.

Μεθόδοι:

- 1)public Player(String name,Color color,Pawn pawn1, Pawn pawn2); //Creates an object of type Player and sets its color, name and pawns.(Constructor)
- 2)public String getName(); //Returns the name of the player.(Accessor)
- 3)public Color getColor(); //Returns the color of the player.(Accessor)
- 4)public boolean isWinner(); //Returns true if player win.(Observer)
- 5)public void changeTurn(); //Changes the turn.(Transformer)
- 6)public boolean isTurn(); //Returns true if it is player's turn to play. (Observer)
- 7)public String toString(); //Returns a string describing the attributes of the player.(Accessor)
- 8)public Pawn getFirstPawn(); //Returns the first pawn of the player. (Accessor)
- 9)public Pawn getSecondPawn(); //Returns the second pawn of the player. (Accessor)

Κλάση Deck

Κλάση για το ταμπλό και τη στοίβα με τις κάρτες.

Πεδία:

- 1)private ArrayList <Card> packOfCards; //Contains all the cards.

2)private ArrayList <Square> squaresOfBoard; //Contains all the squares of the board.

3)private Card latestCardPlayed; //The current card played.

4)private int cardsLeft; //The amount of the cards that haven't been played yet.

Μεθόδοι:

1)public ArrayList <Card> getPackOfCards(); //Returns the pack of the cards.(Accessor)

2)public ArrayList <Square> getBoard(); //Returns the list with the squares of the board.(Accessor)

3)public void initialize(); //Initializes the board with the squares, the deck with the cards and the two players with their two pawns each.(Transformer)

4)public void getCard(); //Selects a card.(Transformer)

5)public Card getLatestCard(); //Returns the latest card played.(Accessor)

6)public int getCardsLeft(); //Returns the amount of the cards that haven't been played yet. (Accessor)

7)public void reshuffling(); //Makes all the cards able to be played again.(Transformer)

- Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Κλάση View

Αυτή η κλάση περιέχει τα γραφικά του παιχνιδιού δηλαδή οτιδήποτε βλέπουμε στο παράθυρο του παιχνιδιού. Πιο συγκεκριμένα θα δημιουργεί ένα frame(JFrame) στο οποίο θα προσθέτει ένα panel(JLayeredPane). Αυτό το panel θα περιέχει το menu (JMenuBar), τα buttons(JButton) fold, packOfCards και currentCard, το background (JLabel) για το ταμπλό του επιτραπέζιου, το tableBackground (JLabel) που θα αναπαριστά την τσόχα/τραπέζι, τα τετράγωνα (JLabel + ειδικό image/color σε μερικές περιπτώσεις), και ένα Info Box(JTextArea) .

Πεδία:

- 1)private JMenuBar menuBar; //The menu of the application.
- 2)private JButton packOfCards; //The button for the pack of the cards.
- 3)private JButton currentCard; //The button for the last card played.
- 4)private JButton foldButton; //The fold button.
- 5)private JButton[] player1Pawns; //The pawns of the first player.
- 6)private JButton[] player2Pawns; //The pawns of the second player.
- 7)private JLabel[] squares; //The squares of the board.
- 8)private JTextArea infoBox; //The info box.
- 9)private JLabel background; //The background of the board.
- 10)private JLayeredPane basic_panel; //The panel.
- 11)private JLabel tableBackground; //The background of the table.

Μεθόδοι:

- 1)public void createFoldButton(); //Creates the fold button.(Transformer)
- 2)public JButton getFoldButton(); //Returns the fold button. (Accessor)
- 3)public void createPackOfCards(); //Creates the packOfCards button.(Transformer)
- 4)public JButton getPackOfCards(); //Returns the packOfCards button. (Accessor)

5)public void createInfoBox(); //Creates the info box.(Transformer)

6)public void updateLastCard(Card c); //Updates the currentCard button image.(Transformer)

7)public void createSquares(ArrayList<Square> squares); //Creates the squares of the board.(Transformer)

8)public void createPawns(); //Creates the pawns of the players.(Transformer)

9)public JButton[] getPlayer1Pawns(); //Returns the pawns of the player 1. (Accessor)

10)public JButton[] getPlayer2Pawns(); //Returns the pawns of the player 2. (Accessor)

11)public void updateInfoBox(String info); //Updates the info on the info box.(Transformer)

12)public void updatePawn(Pawn redPawn1,Pawn redPawn2, Pawn yellowPawn1, Pawn yellowPawn2); //Updates the position of all pawns.(Transformer)

13)public createMenu(); //Creates the menu bar.(Transformer)

14)public void setBackground(); //Sets the background of the board.(Transformer)

15)public void initialize(); //Initializes all the components and adds them to the basic_panel.(Transformer)

16)public View(); //Creates an object of type View(of type JFrame as View extends JFrame) and adds the panel in it. (Constructor)

17)public JTextArea getInfoBox(); //Returns the info box.(Accessor)

18)public void setTableBackground(); //Sets the background of the table.(Transformer)

19)public void startAgain(JButton pawn); //Sends pawn to its start square location.(Transformer)

20) public void reachHome(JButton pawn); //Sends pawn to its home square location. (Transformer)

- Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Κλάση Controller

Η κλάση Controller αποτελεί τον εγκέφαλο του προγράμματος αφού όλες οι λειτουργίες εκτελούνται από αυτήν. Επίσης αποτελεί τον συνδετικό κρίκο μεταξύ των κλάσεων View και Model. Οι δύο αυτές κλάσεις είναι ξένες μεταξύ τους και συνδέονται μόνο μέσω της κλάσης Controller που ελέγχει το παιχνίδι και τη ροή του.

Πεδία:

- 1)private Player player1; //First player.
- 2)private Player player2; //Second player.
- 3)private Player currentPlayer; //The player who's turn to play.
- 4)private Deck cardsNBoard; //The board and the pack of the cards.
- 5)private View view; //The graphical interface for the user.

Εσωτερικές κλάσεις που υλοποιούν το interface MouseListener και μας βοηθάνε ώστε ο χρήστης να αλληλεπιδρά με τη γραφική διεπαφή:

- 1)class FoldListener //Mouse listener για το fold button

και η συνάρτηση της

```
public void mouseClicked(MouseEvent arg0); //Changes the turn if the player can select fold.
```

- 2)class CardsListener //Mouse listener για το packOfCards button

και η συνάρτηση της

```
public void mouseClicked(MouseEvent arg0); //Plays a new card if that can happen.
```

- 3)class PawnListener //Mouse listener για τα pawn buttons.

και η συνάρτηση της

```
public void mouseClicked(MouseEvent arg0); //Moves the pawn if that can happen.
```

4)class MenuListener //Mouse listener για το menu bar.

και η συνάρτηση της

public void mouseClicked(MouseEvent arg0); //if you click on new game it creates a new game.

Και τέλος οι μεθόδοι:

1)public void initialize(); //Initializes the view, the cards and the players.(Transformer)

2)public void setListeners(); //Sets listeners to the fold, cards and pawns buttons.
(Transformer)

3)public boolean checkIfGameFinished(); //Checks if someone won and the game finished.(Observer)

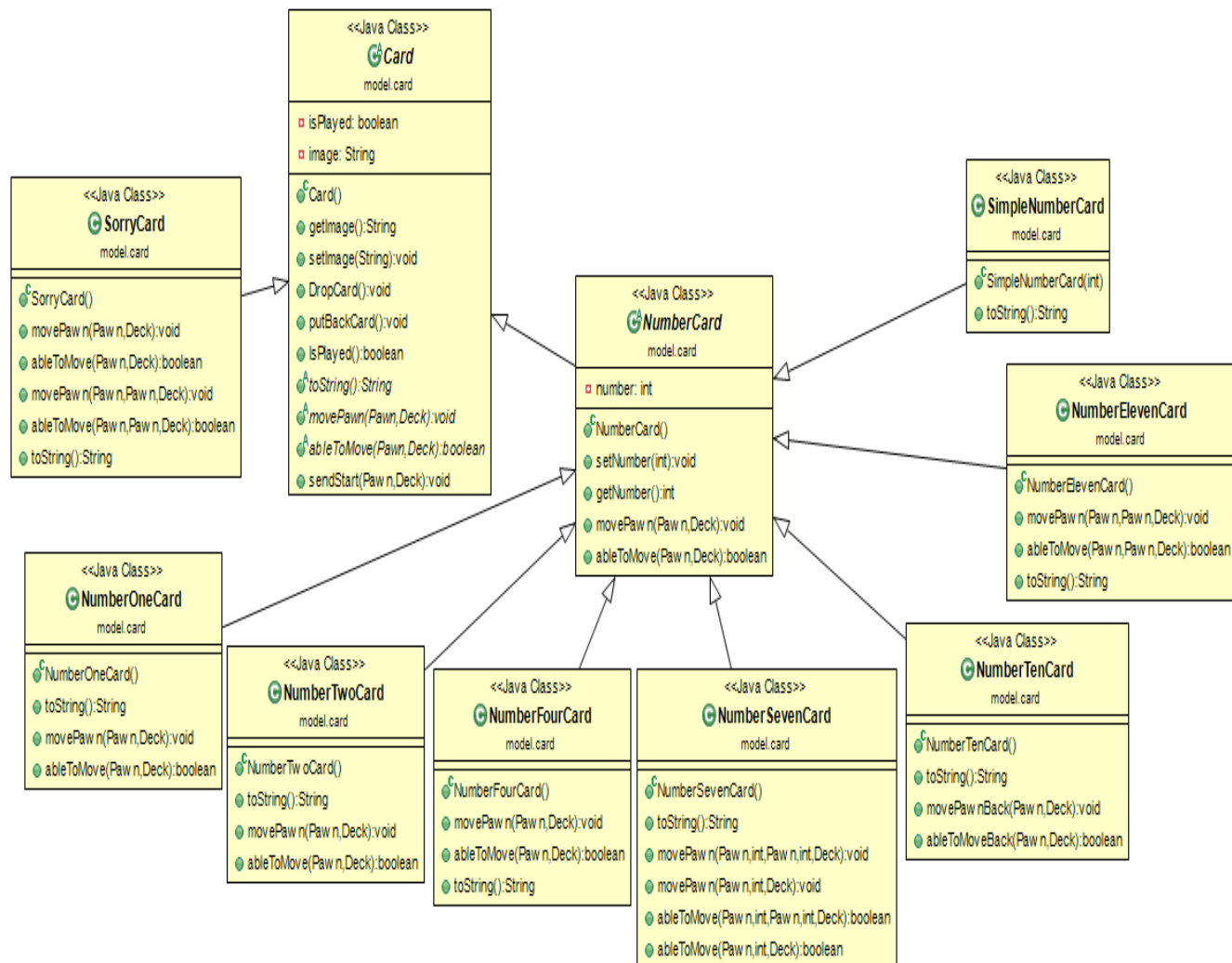
4)public void fold(); //Checks if player can fold and if he can then it changes the turn and the card.(Transformer)

5)public void changeTurn(); //Changes the turn.(Transformer)

6)public void showWinningMessage(String message); //Shows who won the game.(Transformer)

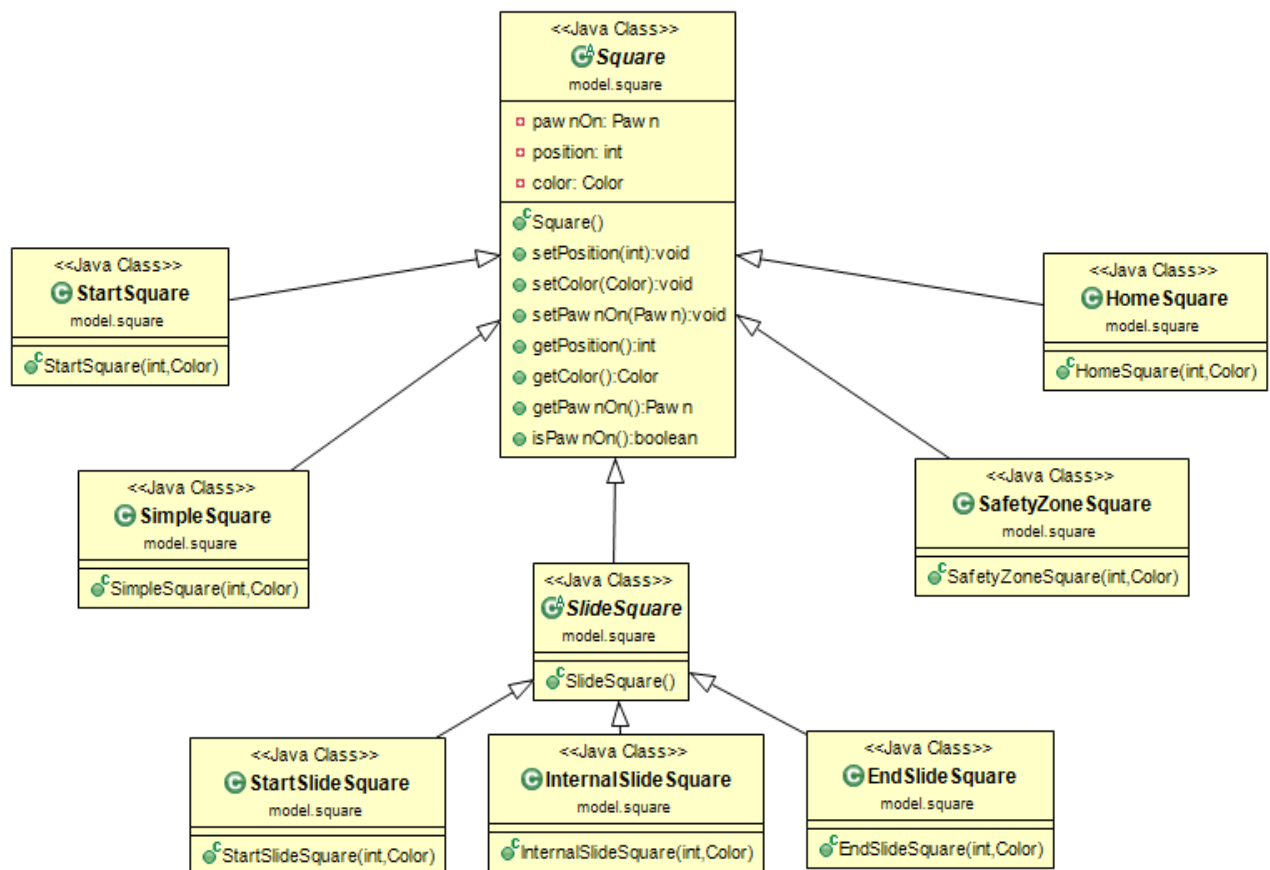
7)public static void newGame(); //Creates a new game(new controller,view,model).
(Transformer)

- Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML
- UML Diagram για τις κάρτες



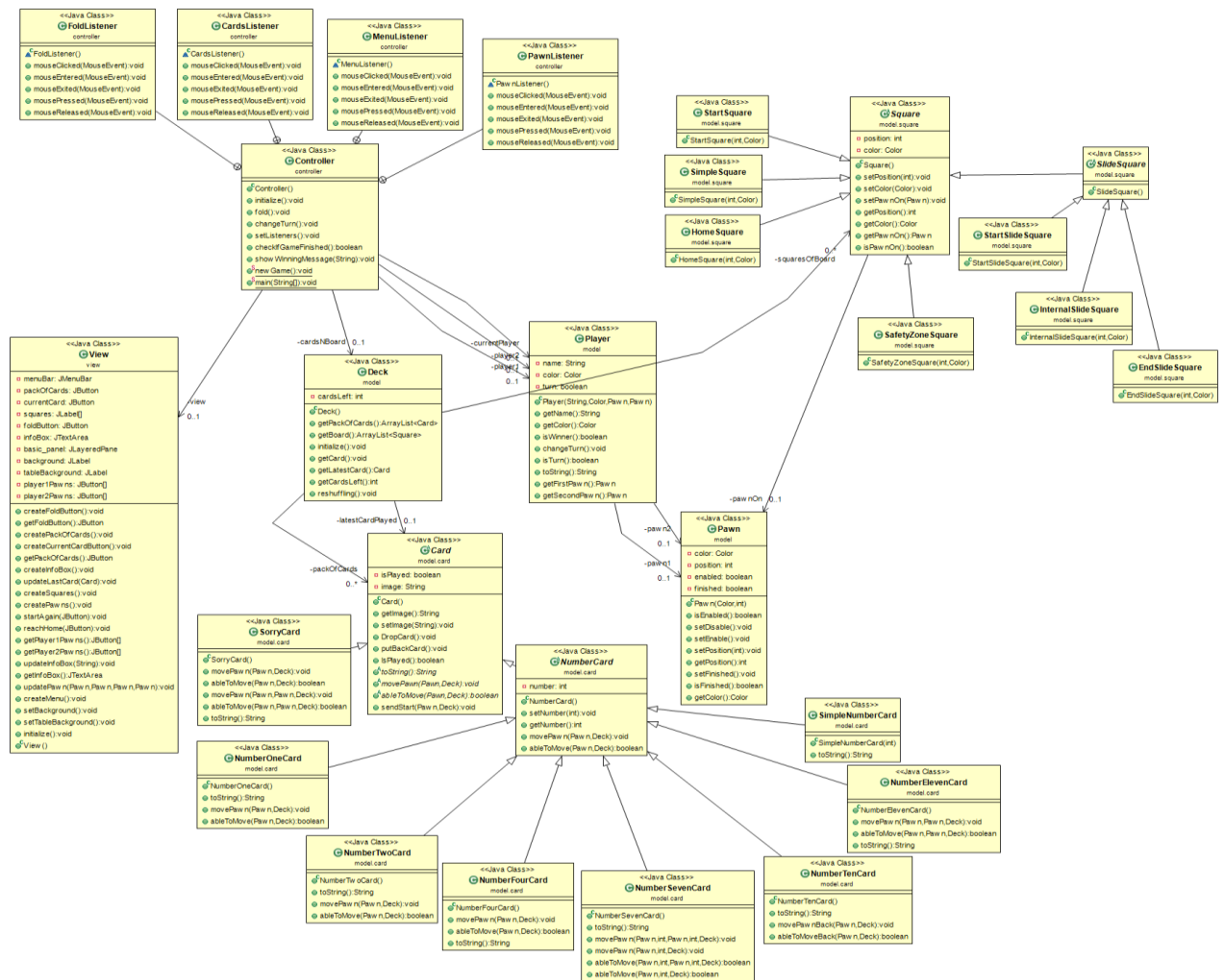
Βλέπουμε ότι η abstract κλάση Card είναι η υπερκλάση όλων των καρτών. Οι αμέσως επόμενες υποκλάσεις της είναι η Sorry Card και η NumberCard. Η NumberCard είναι επίσης abstract και αποτελεί την υπερκλάση των κλάσεων SimpleNumberCard, NumberOneCard, NumberTwoCard, NumberFourCard, NumberSevenCard, NumberTenCard, NumberElevenCard.

- UML Diagram για τα Squares



Η abstract κλάση **Square** αποτελεί την υπερκλάση όλων των τετραγώνων. Οι αμέσως επόμενες υποκλάσεις της είναι οι κλάσεις **StartSquare**, **SafetyZoneSquare**, **HomeSquare**, **SimpleSquare** καθώς και η abstract κλάση **SlideSquare** η οποία αποτελεί υπερκλάση των κλάσεων **StartSlideSquare**, **InternalSlideSquare** και **EndSlideSquare**.

- UML Diagram για όλες τις κλάσεις του MVC



Εδώ φαίνεται το UML διάγραμμα για όλες τις κλάσεις που φτιάξαμε. Αναλύσαμε παραπάνω τις σχέσεις μεταξύ των καρτών καθώς και μεταξύ των τετραγώνων. Το μόνο που προστίθεται είναι ότι η κλάση Square συνδέεται με την κλάση Pawn μέσω του πεδίου της που δείχνει αν υπάρχει κάποιο πιόνι στην ίδια θέση με το τετράγωνο. Εδώ βλέπουμε και τις υπόλοιπες κλάσεις. Η κλάση Deck συνδέεται με την Card μέσω των πεδίων της για το πακέτο με τις κάρτες και για την πιο πρόσφατη κάρτα που έχει παιχτεί. Επίσης συνδέεται με την κλάση Square μέσω του πεδίου της για τα τετράγωνα του ταμπλό.

Ακόμη, η κλάση Player συνδέεται με την κλάση Pawn μέσω των πεδίων της για τα δυο πιόνια που αντιστοιχούν στον κάθε παίκτη. Τέλος, η κλάση Controller συνδέεται με την κλάση View λόγω του πεδίου της για να χειρίζεται τη γραφική διεπαφή του παιχνιδιού, με την κλάση Player μέσω των πεδίων της που αναπαριστούν τον πρώτο και τον δεύτερο παίκτη καθώς και τον παίκτη που είναι η σειρά του να παίξει, και εν τέλει με την κλάση Deck που αναπαριστά το ταμπλό του παιχνιδιού καθώς και το πακέτο με τις κάρτες. Επιπλέον περιέχει τις εσωτερικές κλάσεις FoldListener, CardsListener, PawnListener και MenuListener οι οποίες μας βοηθούν στην διαχείριση και εξέλιξη του παιχνιδιού.

- **Λειτουργικότητα**

Το παιχνίδι παίζεται με δύο παίκτες. Αρχίζει με την αρχικοποίηση του ταμπλό, των καρτών και των παιχτών. Γίνεται τυχαία επιλογή ενός παίχτη για να ξεκινήσει το παιχνίδι. Ο παίχτης κινεί το πιόνι που θέλει πατώντας πάνω στο αντίστοιχο button.

Μπορεί να ξεκινήσει μόνο αν τύχει κάρτα 1 ή 2. Μπορεί επίσης με την κάρτα Sorry αρκεί όμως ο αντίπαλος να έχει ξεκινήσει ήδη κάποιο πιόνι και αυτό το πιόνι να μην είναι σε ζώνη ασφαλείας, αρχική θέση ή σπίτι. Η επιλογή νέας κάρτας γίνεται πατώντας το button με τη στοίβα από κάρτες. Οι κινήσεις είναι εφικτές μόνο αν στη θέση που πρόκειται να πάει το πιόνι δεν υπάρχει πιόνι ίδιου χρώματος (αυτό δεν ισχύει για την αρχική θέση και σπίτι). Αν ο παίχτης δεν μπορεί να κινήσει κάποιο πιόνι του μπορεί να κάνει fold πατώντας το fold button. Εφόσον το πιόνι μπορεί να κάνει μια κίνηση πρέπει να την κάνει (εκτός από την ανταλλαγή στην κάρτα 11 που δεν είναι υποχρεωτική). Αν τύχει κάρτα 1,2,3,5,8,10,11,12 τότε κινείται τόσες θέσεις μπροστά όσες και ο αριθμός. Βέβαια η κάρτα 2 έχει επίσης την ιδιότητα ότι ο παίχτης ξαναπαίζει, η κάρτα 10 ότι το πιόνι μπορεί να πάει 1 θέση πίσω και η κάρτα 11 έχει την ιδιότητα ότι μπορεί να ανταλλάξει το πιόνι με ένα από τα πιόνια του αντιπάλου αρκεί αυτό να μη βρίσκεται σε ζώνη ασφαλείας, αρχική θέση ή σπίτι. Η κάρτα 7 έχει την ιδιότητα ότι μπορεί να κουνήσει και τα δύο πιόνια του παίχτη κατά 7 συνολικά θέσεις. Η κάρτα 4 κινεί το πιόνι 4 θέσεις πίσω. Η κάρτα Sorry ανταλλάσει ένα πιόνι του παίχτη που βρίσκεται στην αρχική του θέση με ένα πιόνι αντιπάλου αρκεί αυτό να μη βρίσκεται σε ζώνη ασφαλείας, αρχική θέση ή σπίτι. Με τη βοήθεια του java dialog μπορεί ο χρήστης να επιλέξει τι κίνηση θέλει να κάνει.

Ένα πιόνι μπορεί να μπει στη ζώνη ασφαλείας του μόνο εφόσον κινείται προς τα εμπρός και όχι προς τα πίσω. Βέβαια μπορεί να κινηθεί προς τα πίσω και την επόμενη φορά που θα παίξει να κινηθεί εμπρός και να μπει στη ζώνη ασφαλείας του. Εάν ένα πιόνι μετακινηθεί σε τετράγωνο στο οποίο βρίσκεται πιόνι διαφορετικού χρώματος τότε θα πάρει αυτό τη θέση του και θα στείλει το αντίπαλο πιόνι στην αρχική του θέση. Επιπλέον, εάν βρεθεί σε αρχική θέση slide διαφορετικού χρώματος τότε τσουλάει μέχρι τη τελική θέση αυτού του slide και στέλνει στις αρχικές τους θέσεις κάθε πιόνι που βρίσκεται μέσα σε αυτό το slide, ακόμα και πιόνια ίδιου χρώματος.

Ένα πιόνι φτάνει στον τελικό προορισμό του δηλαδή στο σπίτι μόνο όταν φέρει ακριβώς τον αριθμό που χρειάζεται (π.χ αν απέχει 5 τετράγωνα από το σπίτι χρειάζεται τη κάρτα 5 για να τερματίσει). Αν φέρει λιγότερο από 5 τότε πρέπει να κινήσει το πιόνι εάν βέβαια είναι εφικτή μια τέτοια κίνηση. Εάν φέρει

περισσότερο από 5 τότε πρέπει να κινήσει αναγκαστικά το άλλο πιόνι του αν φυσικά είναι εφικτό. Σε διαφορετική περίπτωση έχει την επιλογή του fold. Το παιχνίδι τελειώνει μόνο όταν δύο πιόνια του ίδιου χρώματος φτάσουν στον τελικό τους προορισμό και ο παίχτης που τα έχει είναι ο νικητής.

Κάποιες περαιτέρω διευκρινήσεις:

Τα τετράγωνα του ταμπλό έχουν κάποιο συγκεκριμένο position ώστε να κινούνται τα πιόνια πάνω σε αυτά. Τα πάνω άσπρα τετράγωνα έχουν positions από 0 έως 15 (από τα αριστερά προς τα δεξιά) όπου στις θέσεις 1 έως 4 και 9 έως 13 υπάρχουν τα κόκκινα slides. Τα δεξιά άσπρα τετράγωνα έχουν positions από 15 έως 30 (από πάνω προς τα κάτω) όπου στις θέσεις 16 έως 19 και 24 έως 28 υπάρχουν τα μπλε slides. Τα κάτω άσπρα τετράγωνα έχουν positions από 30 έως 45 (από τα δεξιά προς τα αριστερά) όπου στις θέσεις 31 έως 34 και 39 έως 43 υπάρχουν τα κίτρινα slides. Τα αριστερά άσπρα τετράγωνα έχουν positions από 45 έως 59 (από κάτω προς τα πάνω) και το 0 που είναι το ψηλότερο από αυτά τα τετράγωνα. Στις θέσεις 46 έως 49 και 54 έως 58 υπάρχουν τα πράσινα slides. Η κόκκινη ζώνη ασφαλείας έχει positions από 60 έως 64 (από πάνω προς τα κάτω) και το κόκκινο σπίτι έχει position 65. Η κίτρινη ζώνη ασφαλείας έχει positions από 66 έως 70 (από κάτω προς τα πάνω) και το κίτρινο σπίτι έχει position 71. Τέλος, η κόκκινη αρχική θέση έχει position 72 και η κίτρινη αρχική θέση έχει position 73.

Τι δεν επιτεύχθηκε;

Έγιναν όλα όσα ζητήθηκαν με εξαίρεση:

- a) Το τετράγωνο StartSquare όπως φαίνεται στα γραφικά έχει κυανό χρώμα (το χρώμα του background του) και όχι το άσπρο χρώμα όπως στην φωτογραφία.
- b) Η κάρτα 7 δεν λειτουργεί πάντα όπως θα έπρεπε. Συγκεκριμένα, αν δύο πιόνια ίδιου χρώματος απέχουν μικρή απόσταση μεταξύ τους, π.χ το πιόνι 1 βρίσκεται 3 τετράγωνα μπροστά από το πιόνι 2, τότε ο χρήστης δεν μπορεί να επιλέξει να κινήσει το πιόνι 2 κατά 3 και το πιόνι 1 κατά 4 τετράγωνα διότι κατά τον έλεγχο των κινήσεων η συνάρτηση ableToMove βλέπει ότι 3 τετράγωνα μπροστά υπάρχει ένα πιόνι ίδιου χρώματος και επιστρέφει false.
- c) JUnit tests.

ΤΕΛΟΣ ΦΑΣΗΣ Β