PROJECT HY-252 2019-2020

Sorry! Board Game

Project

ΓΙΑΝΝΗΣ ΑΝΤΩΝΟΓΙΑΝΝΑΚΗΣ
ΑΜ:4104

Σχεδιασμός της εργασίας

Η υλοποίηση της εργασιας θα βασιστει πανω στο μοντελο **MVC(**Model View Controller).Έτσι, σκοπός μας είναι ο Controller να είναι ο συνδετικός κρίκος των Model και View.Στη συνέχεια της αναφοράς μας θα αναλύσουμε λίγο ιδιαίτερα τα κομμάτια του Model και του View που ειναι σημαντικότερα για την φάση Α και στο τελος θα αναφερθούμε και στο View.

Package Model

Σε αυτό το πακέτο θα περιέχονται η αφηρημένη κλάση **Card**,οι κλάσεις **Card**, **NumberCard**, **SorryCard** κλάσεις που κληρονομούν την **Card**.Επίσης υπάρχουν οι κλάσεις **NumberElevenCard**,**NumberFourCard**,**NumberOneCard**, **NumberSevenCard**,**NumberTenCard**,**NumberTwoCard** και **SimpleNumberCard** που κληρωνομούν την κλάση **NumberCard**.Επιπλέον,υπάρχουν οι κλάσεις **Deck**, **Pawn,Player**,**Turn** και αφηρημένη κλάση **Square**.Την κλάση **Square** κληρονομούν οι κλάσεις **StartSquare**,**SimpleSquare**,**SafetyZoneSquare**,**HomeSquare** Και **SlideSquare**.Τέλος την κλάση **SlideSquare** κληρονομούν οι κλάσεις **StartSlideSquare**,**InternalSlideSquare** και **EndSlideSquare**.

Abstract class Card and other Classes for Cards

Αρχικά, φτιάχνοντας την αφηρημένη κλάση **Card** μπορουμε να προσπελάσουμε τα δεδομένα χωρίς να πρέπει να ορίσουμε αν μια καρτά ειναι ειδική η απλή..

Τα attributes:

1) private int moves; //The moves from the card.
2) private boolean isThrown; //if a card is thrown.

Οι μέθοδοι:

1. public void setDescription(String des){} **Transformer (Mutative)**
   Sets the Description of a Card.

2. public String getDescription(){} **Accessor (Selector)**
   Returns the Description of a Card.

3. public void setThrow(boolean t){} **Transformer (Mutative)**
   Sets if the card is thrown.

4. public boolean getThrow(){} **Accessor (Selector)**
   Returns if the card is thrown.

5. public void setMoves(int i){} **Transformer (Mutative)**
   Sets the moves of a player.

6. public int getMoves(){} **Accessor (Selector)**
   Returns the moves of a player.

7. public void movePawn(Pawn p, Board b){}
   Moves the pawn on the board.

Στην συνέχεια έχουμε την **NumberCard** και **SorryCard** που κάνουν extend την κλάση **Card**.

Class NumberCard

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

Τα attributes:

1)private int value; //The value of a card.

Οι υπόλοιπες μέθοδοι:

1.public NumberCard(){} **Constructor**
Constructs an instance of the class **NumberCard.**

Class SorryCard

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

Τα attributes:

1)private Pawn p; //take one pawn from the other player for swap;

2) private boolean isThrown; //if a card is thrown

Οι μέθοδοι:

1.public SorryCard(){}
Constructs a new instance of SorryCard

2.public void checkSwap(Pawn p1, Board b)
checks if can be done the swap.

Class NumberElevenCard

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

Οι μέθοδοι:

1.public NumberElevenCard(){}
Constructs a new instance of NumberelevenCard.

2.public void CheckSwap(Pawn p, Pawn p2, Board b)()
Checks if swap can be done.

Class NumberFourCard

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

Οι μέθοδοι:

1.public NumberFourCard(){}
Constructs a new instance of NumberFourCard.

2.public void movePawn(Pawn p, Board b){}
Moves the pawn on the board.

Class NumberOneCard

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

Τα attributes:

1)private boolean inStart;//if a pawn is in the start square

Οι μέθοδοι:

1.public NumberOneCard(){}
Constructs a new instance of NumberOneCard.

2.public void setStart(boolean start){} **Transformer (Mutative)**
Sets if a pawn is on the startSquare

3.public boolean getStart(){} **Accessor (Selector)**
Return if a pawn is on the startSquare

4.public void movePawn(Pawn p, Board b)
moves a pawn on the board.

Class NumberSevenCard

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

Οι μέθοδοι:

1. public NumberSevenCard(){}
   Constructs a new instance of NumberSevenCard

2. public void movePawn(Pawn p1, Pawn p2, int a, int b){}
   Move a pawn on the Board.

Class NumberTenCard

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

Οι μέθοδοι:

1.public NumberTenCard(){}
Constructs a new instance of NumberTenCard.

2.public void movePawn(Pawn p, Board b)
moves a pawn on the board.

Class NumberTwoCard

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

Οι μέθοδοι:

1. public voidNumberTwoCard(){}
   Constructs a new instance of NumberTwoCard.
2. public void movePawn(Pawn p, Board b){}
   moves a pawn on the board.

Class SimpleNumberCard

Οι μέθοδοι:

1. public SimpleNumberCard(){}
   Constructs a new instance of SimpleNumberCard.
2. Public void movePawn(Pawn p, Board b, int moves){}
   moves a pawn on the board.

Abstract Class Square and other Classes for Squares

Η αφηρημένη κλάση **Square** μας παρέχει τις εξής μεθόδους:

1. private void setColor(int color){} **Transformer (Mutative)**
   Sets the color of a Square.
2. private int getColor(){} **Accessor (Selector)**
   Return the color of a Square.
3. public void set_has_Pawn(boolean p){} **Transformer (Mutative)**
   Sets pawn on a Square.
4. public boolean get_has_Pawn(){} **Accessor (Selector)**
   Return if a Square has pawn.

Στην συνέχεια έχουμε τις κλάσεις **StartSquare,SimpleSquare,SlideSquare, SafetyZoneSquare, HomeSquare** που κάνουν **extend** την κλάση **Square.**

Class StartSquare

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

Τα attributes:

1) private boolean hasPawn; //if a Square has pawn
2) private int color; //the color of the Square

Οι μέθοδοι:

1. public StartSquare(){}
   Constructs a new instance of StartSquare.

Class SimpleSquare

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

Τα attributes:

1)private boolean hasPawn; //if a Square has pawn.
2)private int color; //the color of a Square.

Οι μέθοδοι:

1. public SimpleSquare(){}
   Constructs a new instance of SimpleSquare.

Class SafetyZoneSquare

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

Τα attributes:

1. private boolean hasPawn; //if a square has pawn
2. private int color; // the color of the Square

Οι μέθοδοι:

1. public SafetyZoneSquare
   Constructs a new instance of SafetyZoneSquare

Class HomeSquare

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

Τα attributes:

1) private boolean hasPawn; //if a Square has pawn
2) private int color; // the color of the Square

Οι μέθοδοι:

1. public HomeSquare(){}
   Constructs a new instance of HomeSquare.

Class SlideSquare

Εδώ θα αναφέρουμε τα attributes και τις υπόλοιπες μεθόδους που έχει η κλάση αυτή.Επιπλέον, οι κλάσεις **StartSlideSquare,InternalSlideSquare** και **EndSlideSquare** που κάνουν **extend** την κλάση **SlideSquare.**

Τα attributes:

1)private boolean hasPawn; //if a Square has pawn
2)private int color; // the color of a Square

Οι μέθοδοι:

1. public SlideSquare(){}
   Constructs a new instance of SlideSquare.

Class StartSlideSquare

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

Τα attributes:

1) private boolean hasPawn; //if a Square has pawn
2) private int color; //the color of a Square

Οι μέθοδοι:

1. public StartSlideSquare(){}
   Constructs a new instance of StartSlideSquare.

Class InternalSlideSquare

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

Τα attributes:

1. private boolean hasPawn; //if a Square has pawn
2. private int color; //the color of a Square

Οι μέθοδοι:

1. public InternalSlideSquare(){}
   Constructs a new instance of InternalSlideSquare.

Class EndSlideSquare

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

Τα attributes:

1) private boolean hasPawn; //if a Square has pawn
2) private int color; //the color of a Square.

Οι μέθοδοι:

1. public EndSlideSquare(){}
   Constructs a new instance of EndSlideSquare.

Class Deck

Η κλάση αυτή είναι υπεύθυνη για να δημιουργεί τις κάρτες να τις ανακατέυει όταν τελειώσουν,να αρχικοποιεί το ταμπλό και να δίνει στους παίκτες νέες κάρτες.Τέλος να ελέγχει αν ο παίκτης μπορεί να πάει πάσο.Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private ArrayList<Card> cards; //Make a list of cards

Οι μέθοδοι:

1. public void init_Board(){}
   Initialize the board and pawns
2. public Deck() {}
   constructs a new instance of Deck.
3. public void shuffle Cards(){}
   Shuffles the cards when the list is empty.
4. public void isEmpty(){}
   check if the list of cards is empty.
5. public void removeCard(Card i){}
   removes a card from a list.
6. public Card getCard(int index){}
   get a card from a list.
7. Public int size(){}
   get the size of cards list.
8. Public ArrayList<Card> getCards(){}

get all the Cards

9. public void checkFold(){}
   checks if a player can press fold.

Class Player

Η κλάση αυτή είναι υπεύθυνη να δημιουργεί παίκτες.Ο κάθε παίκτης έχει ένα μοναδικό χρώμα,όνομα και έχει δύο πιόνια που πρέπει να πάει στο Home ώστε να κερδίσει το παιχνίδι.Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private String name; //The name of a player
2. private int choice, ID; //The ID of a player and his choice in special cards
3. private boolean hasPlayed,finished; //if a player finish a round,and if player finished the game.
4. Private int color; //the color of a player
5. private ArrayList<Pawn> pawns; //the pawns of a player

Οι μέθοδοι:

1. public Player(){}
   Constructs a new instance of Player
2. public int getID(){} **Accessor(selector)**
   Returns the id of a player
3. public void setID(){} **transformer(mutative)**
   Sets the ID for a player

4. public String getName(){} **Accessor(selector)**
   Returns the name of a player
5. public int getChoice(){} **Accessor(selector)**
   Returns the choice of a player
6. public void Played(){} **transformer(mutative)**
   Sets if a player has played his turn
7. public void has_finished(){} **transformer(mutative)**
   Sets if a player has finished the game
8. public boolean get_has_finished(){} **Accessor(selector)**
   Returns if a player has finished the game

Class Pawn

Η κλάση αυτή είναι υπεύθυνη να δημιουργεί πιόνια.Κάθε πιόνι έχει ένα χρώμα και βρίσκεται σε ένα συγκεκριμένο τετράγωνο ενώ επίσης μπορεί να έχει τερματίσει η να είναι ενεργό ακόμα μέσα στο παιχνίδι.Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private int StartX,StartY; //the positions of the startSquare
2. private int posX,posY; //The current positions after start
3. private int col; //the color of a pawn
4. private boolean finish; //if a pawn is on HomeSquare

Οι μέθοδοι:

1) public Pawn(){}
   Constructs a new instance of pawn
2) public void setX(){} **transformer(mutative)**
   Sets the x position of a pawn

3) public int getX(){} **accessor(selector)**
   Returns the X position of a pawn
4) public void setY(){} **transformer(mutative)**
   Sets the position Y of a pawn.
5) Public int getY(){} **accessor(selector)**
   Returns the position Y of a pawn

Class Turn

Η κλάση αυτή διαχειρίζεται τη σειρά των παικτών μέσα στο παιχνίδι.Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1.private int currentID //the id of the player who has turn.

Οι μέθοδοι:

1. public void NumberOfPawns
   Number of active pawns of a player
2. public getID(){} **Accessor(Selector)**
   Returns the ID of a player.
3. Public Turn(){}
   Constructs a new instance of Turn.
4. Public void setID(){}
   Sets the ID for the player who has turn

Package Controller

Class Controller

Αυτή η κλάση αποτελεί τον εγκέφαλο του παιχνιδιού.Είναι υπεύθυνη για την ομαλή λειτουργία του παιχνιδιού.Επίσης ελέγχει την σειρά που θα παίξουν οι παίκτες,τον τερματισμό του παιχνιδιού και την ανάδειξη του νικητή και τέλος συνδέει το **view** με το **Model.**Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η κλάση αυτή.

Τα attributes:

1. private int fold; //increases when the player press fold
2. private ArrayList<Player> players; //List with instances of player
3. private Player p1,p2; //the two players
4. private Turn turn; //make an instance of Turn
5. private Deck allcards // make an instance of Deck

Οι μέθοδοι:

1. public Controller(){}
   Constructs a new instance of Controller
2. public void set_Fold(){} **transformer(mutative)**
   increases the fold variable +1
3. public void new_Game(){}
   starts a new game
4. public boolean GameHasfinished(){} **Observer**
   Returns if a game has finished
5. public void seeTurn(){} **Accessor(selector**)
   check who player is playing now

6. public void set_Turn(){}
   Set who player will play

Package View

Σε αυτο θα περιεχονται τα γραφικα του παιχνιδιου.Αρχικα φτιαχνουμε το ταμπλο τα κουμπια τα τετραγωνα οι καρτες τα πιονια.Αναλυτικοτερα:

Τα attributes:

7 Jbuttons: Που αντιστοιχούν 2 για τι κάρτες 4 για τα πιόνια 1 για το fold κουμπί.

74 Jlabels που χρησιμοποιούνται για το ταμπλό.

1 JtextArea που εμφανίζει πληροφορίες όπως πχ ποιος παίκτης παίζει τώρα και πόσες κάρτες εχουν απομείνει.

Οι μεθόδοι:

1.private void initComponets() {}
//initialize the componets for the board

2. public void init_Buttons() {}
//initialize the buttons for the board

3. public void actionPerformed(ActionEvent e){}
   //what to do when the BackCardButton is pressed

4. private class Fold implements ActionListener{}
//what to do when the fold button is pressed

5. public static void music()
//Playing some nice music in the Background

Σχετικά με την υλοποίηση μου:

Αρχικά, έχω προσθέσει μερικά σχολια σε σχέση με την Α φάση.Αρχικοποιώ το ταμπλό με τα πιονιά τις καρτες και τους παίκτες.Επισης υπάρχει η τήρηση σειράς λειτουργούν οι κάρτες 1,2,3,4,5,8,12 κομπλέ(μόνο την κάρτα Sorry δεν εχω φτιάξει και επίσης η 11 και η 7 κάνουν κινήσεις με το αντιστοίχο νούμερο που εχουν).Υπάρχουν κανόνες για εάν υπάρχει πιόνι σε ενα κουτί δεν γίνεται να πάει άλλο (ο έλεγχος γίνεται με μία boolean μεταβλητή οπου αρχικά ελέγχει αν ειναι false το βάζει εκεί αλλιώς αν ειναι true το παιχνίδι συνεχίζεται κανονικά.Αν ένα πιόνι φτάσει στην αρχή μιας τσουλήθρας που έχουν διαφορετικό χρώμα τότε κυλάει μέχρι το τέλος της.Το κουμπί Fold λειτούργει μόνο για να αλλάζει σειρά χωρίς όμως να γίνεται έλεγχος.Τέλος,σε μερικά πράγματα δεν πάω σύμφωνα με την εκφώνηση όπως για παράδειγμα στις κάρτες αντί για στιγμιότυπο του Board εχω φτιάξει έναν πίνακα με στιγμιοτύπα των Square οπου βαζω τις συντεταγμένες και το χρώμα που είναι το κάθε τετράγωνο για να ξέρει ο παίκτης που θα καταλήξει το πιόνι του μετά απο το παίξιμο μιας κάρτας.

Τι δεν έχω κάνει:

Δεν έχω κάνει **Έλεγχος για Fold, Σωστή χρήση θέσης Home- Νικητής, JUnit Tests ,Χρήση Κάρτας Sorry! (+ έλεγχος για Safety Zone).**

ΤΕΛΟΣ