

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2020-2021**

[project hy252]

Εισαγωγή

[Σαμπάνι Ντένις]

[csd4739]

[2/1/24]

Think and describe what you plan to do and why it will be useful.

Περιεχόμενα

[1. Εισαγωγή 1](#_Toc530045458)

[2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 1](#_Toc530045459)

[3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller 1](#_Toc530045460)

[4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View 2](#_Toc530045461)

[5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 2](#_Toc530045462)

[6. Λειτουργικότητα (Β Φάση) 2](#_Toc530045463)

[7. Συμπεράσματα 2](#_Toc530045464)

## Εισαγωγή

Το **Model–view–controller** (σε συντομογραφία αναφέρεται ως **MVC**) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για τη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο *Model* το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο *View* χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο *Controller* ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο *Model* και στο *View*.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Σε αυτο το πακετο θα περιεχετε,η διεπεφη Card, οι κλασεις   
NumberCard,NumberElevenCard,NumberFourCard,

NumberOneCard,NumberSevenCard,NumberTenCard,

NumberTwoCard,SimpleNumberCard,SorryCard

Παμε να αναλυσουμε αυτες πρωτα:

Διεπαφη Card:  
Φτιαχνοντας την διεπαφη αυτη μας δινεται η δυνατοτητα να προσπελασουμε τα δεδομενα χωρις να πρεπει να ορισουμε αν μια καρτα ειναι απλη η ειδικη

Η Διεπαφη αυτη περιεχει:

1. Public int getValue();  
   Returns the value of a card
2. Public void setValue(int value);  
   Sets the value of a card
3. Public String getImage();  
   Returns the image name associated with the card
4. Public void setImage();

Sets the image name associated with the card  
  
Class NumberCard:

Attributes

1. private int value;
2. Private String imageName;

Methods:

1. Constructor :

NumberCard(int value){this.value=value};  
Create a number card and initialize its value

1. Public int getValue(){}->@override
2. Public int setValue(int value){}->@override
3. Public getImage(){}->@override
4. Public setImage(Image image){}->@ovveride

Class NumberElevenCard:

1.NumberElevenCard(){super(11,image location);}  
Constructor that calls super constructor to create this type of card

Class NumberFourCard:

NumberFourCard(){super(-4,image location);}  
Constructor that calls super constructor to create this type of card

Class NumberOneCard:

1.NumberOneCard(){super(1,image location);}  
Constructor that calls super constructor to create this type of card  
  
Class NumberSevenCard:

1.NumberSevenCard(){super(7,image location);}  
Constructor that calls super constructor to create this type of card  
  
Class NumberTenCard:

1.NumberTenCard(){super(10,image location);}  
Constructor that calls super constructor to create this type of card

Class NumberTwoCard:

NumberTwoCard(){super(2,image location);}  
Constructor that calls super constructor to create this type of card  
  
Class SimpleNumberCard:

This class is about cards 3,5,8,12

Methods:

1.SimpleNumberCard(int value){super(value,image location);}  
Constructor that calls super constructor to create this type of card

Class SorryCard:

SorryCard(){super(6,image location);}  
Constructor that calls super constructor to create this type of card

Class Deck:

Αυτη η κλαση χρησιμοποιηται για την δημιουργια και λειτουργια μιας τραπουλας  
Attributes:  
 ArrayList<Card> cards;a collection of cards  
  
Methods:

Public void init\_cards(){};  
Method that initializes and shuffles the 44 cards

Setters and getters for the Cards

Public void setCards(ArrayList<Card> cards)  
Public ArrayList<Card> getCards()

Class Pawn:  
Αυτη η κλαση χρησιμοποιηται για την δημιουργια και λειτουργια  
των πιονιων  
Attributes:

1. Private Color color;
2. Private String name;

Methods;Pawn(String name,Player player,Color color){  
 this.name=name;

Class Player:   
Αυτη η κλαση χρησιμοποηται για την δημιουργια παιχτων  
Atrributes:  
private String name;  
private PlayerColor color;  
private int ID;  
  
methods:

1. public Player(String name,PlayerColor int ID){  
    this.name=name;  
    this.color=color;  
    this.ID=ID;}  
   Contructor to create players
2. Public void setPawns(){}  
   method to set pawns to players  
   3.public void setName(String name){this.name=name;}  
   sets the name of the player

4.public String getName(){return name;}  
return players name  
5.public PlayerColor getColor(){return color;}  
return players color  
6.public void setColor(PlayerColor color){this.color=color;}  
sets players color  
  
public enum PlayerColor{RED,YELLOW};  
enum to get the colors we want

Class Turn:  
αυτη η κλαση χρησιμοποιηται για να δουμε ποιανου παιχτη σειρα ειναι  
  
Methods

1. public int GetTurn(){}  
   return who plays now

Public void setTurn(){}  
sets the turn right now

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller η κλαση controller ελεγχει ολο το προγραμμα επικοινωνει και το model και με το view για να καταφερει να λειτουργησει το προγραμμα methods: 1.public static void main(){} creates an instance of controller which gets the game going 2.public void setCards(){} method to initialize the cards 3.public void setFold(){} method if fold button pressed to change turn 4.public int seeTurn(){} method to see the current turn public void initialize()

## Method that intiliazes players and calls other methods to start the game also sets and buttons and the the listeners for them It’s the first method that is being called after the constructor is created,also initializes the info box

## Public void setListeners() sets Listeners for the buttons

## Public void setPawns() sets the pawns for each player

## Public void update current This method drawns a new card ,updates the game with help of another function,updates the turn and the info box and ends the game if it really ended

## Public class DrawListener Method that every time we click on the cards a new one is drawn

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View Η κλαση View ειναι υπευθυνη για το visual part του project η κλαση αυτη περιεχει:

## methods: 1. public View(){} constructor that creates a new window 2.public void initComponents(){} method that initializes pawns and background images of the table 3.public init\_squares(){} method that creates the windows for the game to move the pawns in there

## Public void init\_buttons() method that initializes the buttons of the game

## public void updateCurrentCard method that updates the current card every time we draw a new card

## public void updatePawn(Player player, int value) The MOST important method of the game.It implements the logic of the game.It controls the moves of the pawns and decides when to move accordingly to the card.

## public void setDrawCard8(boolean drawCard8)

## method to set if we need to draw a new card,after drawing card 8

## 7.public boolean isDrawCard8()

## method to get the info that we need to draw a new card,after drawing card 8 8.public void setDrawCard12(boolean drawCard12) method to set if we need to draw a new card,after drawing card 12 9.public boolean isDrawCard12() method to get the info that we need to draw a new card,after drawing card 12 10.public void gameFIN() method to see if the game has ended Then we have slideCheckers for each color Their job is to check the slide rules of the game and move pawns accordingly 11.public void init\_pawns method that initializes pawns 12.public void updateInfobox() updates the info box 13.public void showWinningMessage() method that shows the winning message if any player won

## 

## Λειτουργικότητα (Β Φάση)

Σε αυτήν την ενότητα θα γράψετε στη Β φάση ποια ερωτήματα καταφέρατε να υλοποιήσετε είτε επιτυχώς είτε εν μέρει (και ενδεχομένως ποια όχι).

Στην β φαση εγινε η γενικη υλοποιηση του παιχνιδιου  
Το δυσκολο κομματι για μενα δεν ηταν η λογικη του παιχνιδου αλλα καποια πολυ ιδιαιτερα edge cases.Οι περιπτωσεις που ακομα δεν δουλεουν 100/100 ειναι οι καρτες (3,5,7,11,12) καθως υπαρχουν περιπτωσεις που εχει τερματησει το πιονι αλλα το κουνανε απο την θεση αυτη.Κατα τα αλλα η λογικη εχει ακολουθηθει ακριβως και το παιχνιδι δουλευει σωστα.Δεν καταλαβα τη χρηση του fold κουμπιου γιυατο και δεν το χρησιμοποιησα,αλλα χρησιμοποιησα user input για τα cases που ζηταω κατι απτο χρηστη και αλλαζα την σειρα ο ιδιος.