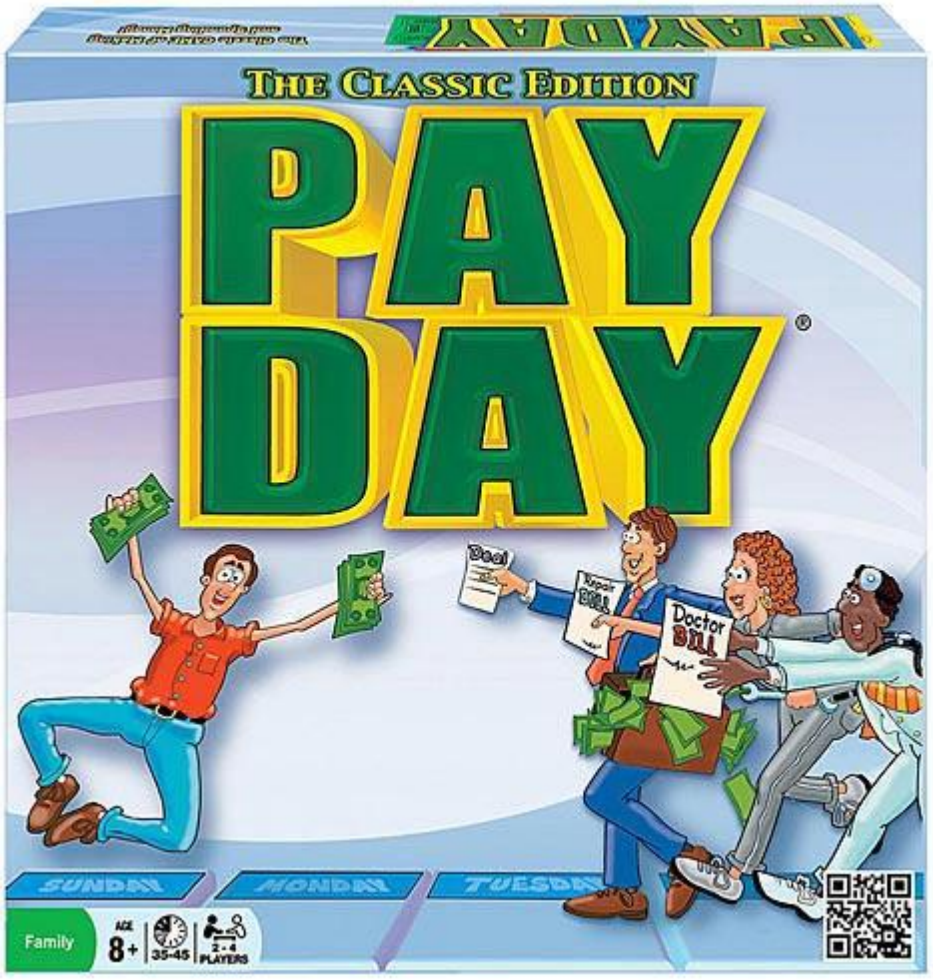
# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΥ-252

## PROJECT ΧΕΙΜ-2016




ΠΑΠΟΥΤΣΑΚΗΣ ΝΙΚΟΛΑΟΣ CSD3035

# Περιεχόμενα

## Σχεδιασμός της Εργασίας

Η εργασία θα υλοποιηθεί σύμφωνα με το μοντέλο MVC (Model View Controller). Σκοπός μας είναι ο controller να συνδέει τα model & view. Στην συνέχεια της αναφοράς θα αναλύσουμε λίγο τα κομμάτια των model & controller που είναι κυριότερα για αυτή την φάση και θα αναφερθούμε λίγο και στο view.

# Package Model

Σ αυτό το πακέτο θα περιέχονται όλες οι διεπαφές και κλάσεις που αφορούν το ταμπλό και τους παίκτες , όπως οι Player , Board , Dice , Jackpot αλλά και οι θέσεις του ταμπλό που κληρονομούν την Board , όπως Charity , Payday κτλ.

## Table Interface & άλλες κλάσεις για το Table

Αρχικά φτιάχνουμε την διεπαφή Table για να ομαδοποιήσουμε κάπως τα περιεχόμενα του ταμπλώ του παιχνιδιού.

Η διεπαφή αυτή μας παρέχει την εξής μέθοδο:

- Public int getMonth(); Accessor
  Get the month players are

Στην συνέχεια έχουμε το Board , Player

## Class Player

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

Attributes:

- private int Money; //  Player's money
- private int Loan; // Loan player has taken
- private int billDept; // Sum of bills player has drawn
- private Vector<Card> dealStack; // All of the deal cards player has bought
- private int Month; // Month that player is playing
- private int playerNo; // Player number
- private static int count = 0; // Counter to count player's constructed
- private boolean Turn; // True if player's turn , false otherwise
- private int position; // Player's position on board
- private boolean finished; // True if player finished game , false otherwise
- private Dice myDice; // Player's dice
- private Player adversary; // Player's opponent
- private boolean gotL; // True if player got Loan , false otherwise

Methods:

- public int GetLoan(); accessor
  Get player's loan
- public Player GetOpponent(); accessor
  Get player's opponent
- public Boolean getGL(); accessor
  Get if player got loan
- public void setGL(boolean b); transformer
  Set if player got loan
- public void SetOpponent(Player p); transformer
  Set the player's opponent
- public void SetDealCards(Card d); transformer
  Add new deal card to player's deal cards
- public Vector<Card> GetDealCards(); accessor
  Get all of the player's  deal cards
- public int getMonth(); accessor
  Get which month player is  playing
- public void setMonth(int x); transformer
  Set which month player is playing
- public void setLoan(int l); transformer
  Add new the new loan amount to player's loan
- public void setMoney(int m); transformer
  Add money to player's money

- public int getMoney(); accessor
  Get player's money
- public void setPosition(int p); transformer
  Set player's position on board
- public int getPosition(); accessor
  Get player's position on board
- public void setbillDept(int x); transformer
  Add new bill amount to player's bills
- public int getbillDept(); accessor
  Get player's bills amount
- public String getName(); accessor
  Get player's name
- public void setTurn(boolean b); transformer
  Set if it is player's turn or not
- public boolean getTurn(); accessor
  Get if it is player's turn or not
- public Dice getDice(); accessor
  Get player's dice
- public void setfinished(boolean b); transformer
  Set true if player finished the game , false otherwise
- public Boolean finished(); accessor
  Get if the player has finished the game

## Class Board

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

Attributes:

- private BoardPosition[] days; //Array with the positions
- private int PlayingMonths; // Playing months choosed at the beginning
- private int month; // Which month players are
- private int Messagethesis = 8; //8 Message card places on board
- private int Dealthesis = 5; //5 Deal card places on board
- private int Sweepthesis = 2; //2 Sweepstakes places on board
- private int Lotterythesis = 3; //3 Lottery places on board
- private int Radiothesis = 2; //2 Radio places on board
- private int Buyerthesis = 6; //6 Buyer places on board
- private int Kazinothesis = 2; //2 kazino places on board
- private int Yardthesis = 2; //2 Yard sales places on board
- private BoardPosition[] shufflearray; // Additional array to shuffle positions

Methods:

- public BoardPosition[] getDays(); accessor
  Get the array with the board positions
- public boolean Sunday(int x); accessor
  Return if it is sunday
- public void InitializeBoard(); transformer
  Initialize the board with the positions(thesis)
- public int getMonth(); accessor
  Get the month players are
- public void setMonth(int m); transformer
  Set the month players are
- public void setPlayingMonths(int m); transformer
  Set the desired months to be played
- public int getplayingMonths(); accessor
  Get the desired months to be played

## Card Interface & άλλες κλάσεις για την Card

Αρχικά είναι η διεπαφή Card η οποία ομαδοποιεί τις κάρτες στο παιχνίδι. Δεν έχει κάποια μέθοδο. Διαχωρίζεται σε δύο κατηγορίες Dealcards & MessageCard, όπως επίσης και την CardStack.

### class Dealcards

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

Attributes:

- private int BuyPrice; // Cards buy value
- private int SellPrice; // Cards sell value
- private String image; // Cards image path
- private String info; // Cards information text
- private String choice; // Cards choice text
- private String choice1; // Cards other choice text

Method:

- public String getMessage(); accessor
  Get the information text
- public String getImage(); accessor
  Get the image path
- public String getChoice(); accessor
  Get the choice text
- public String getChoice1(); accessor
  Get the other choice text
- public int getBuyPrice(); accessor
  Get the buy value of the card
- public int getSellPrice(); accessor
  Get the sell value of the card

## class CardStack

Εδώ θα αναφέρουμε τα attributes και τιε μεθόδους που έχει η κλάση αυτή

Attributes:

- public ClassLoader cldr;
- private Stack<Card> stack; // Stack of the cards
- private Stack<Card> init_stack; // Stack of the rejected cards
- private String[][] deal; // String array for the information of deal cards
- private String[][] mail; // String array for the information of message cards

Method:

- public void InitializeCards(); transformer
  Initialize the card stack
- public Card DrawCard(); accessor
  Get the top card of the stack
- public boolean isEmpty(); accessor
  Check if the card stack is empty
- public void ShuffleStack(); transformer
  Shuffle the cards in stack

## class MessageCard

Εδώ θα αναφέρουμε τις μεθόδους που έχει η κλάση αυτή

Attributes:

- private String image; // Card's image path
- private int amount; // Card's amount
- private String info; // Card's information
- private String choice; // Card's choice text

Method:

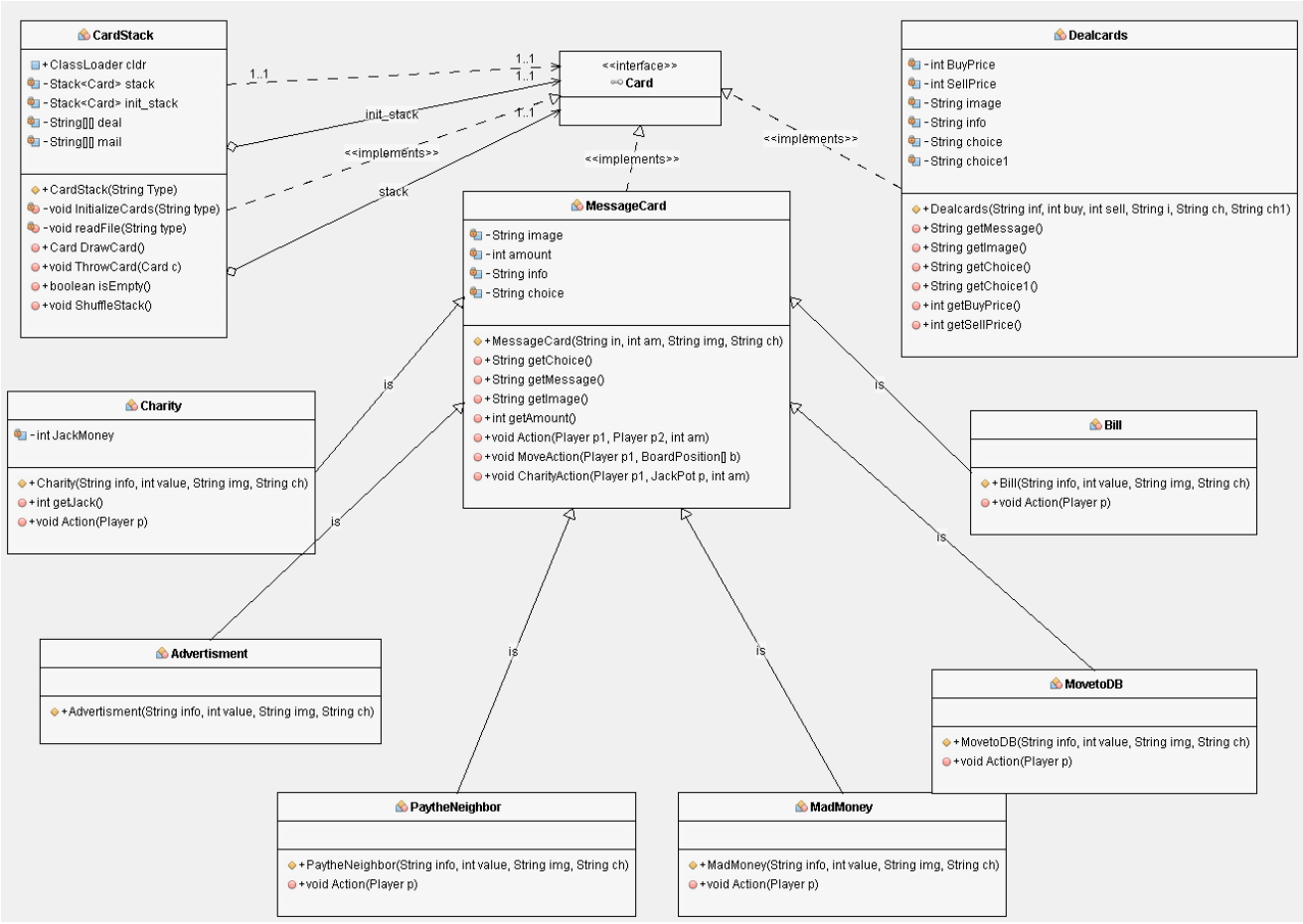
- public String getChoice(); accessor
  Get the choice text
- public String getMessage(); accessor
  Get the information text
- public String getImage(); accessor
  Get the image path
- public int getAmount(); accessor
  Get the amount
- public void Action(Player p1, Player p2,int am); transformer
  According the parameters , will make the necessary actions
- public void MoveAction(Player p1, BoardPosition[] b); transformer
  Change player's position to the nearest Deal or Buyer position
- public void CarityAction(Player p1,JackPot p , int am); transformer
  Take the money from the player and add them to jackpot

Κλάσεις που κληρονομούν την MessageCard

- MadMoney
- Charity
- MovetoDB
- Bill
- PaytheNeighbor
- Advertisment

Μόνο η κλάση Charity έχει attribute , το int JackMoney
Η κλάση Advertisment δεν εκτελεί τίποτα,
η κλάση Charity χρησιμοποιεί την CharityAction της MessageCard ,
όπως η κλάση MovetoDB χρησιμοποιεί την MoveAction της MessageCard ,
ενώ οι υπόλοιπες χρησιμοποιούν την Action της MessageCard.

Εδώ μια αναπαράσταση σε UML της διεπαφής Class , των Dealcards , CardStack & MessageCard με τις υποκλάσεις της.



## Abstract Class BoardPosition & οι υποκλάσεις της

Αρχικά έχουμε την BoardPosition η οποία ομαδοποιεί τις κλάσεις που έχουν σχέση με τις θέσεις στο παιχνίδι και τους κληρονομεί τις εξής μεθόδους.

Attributes:

- private String Image; // Image path of the position

Methods:

- public void toDo(); transformer
  Do the right thing according the position
- public void setImage(String i); transformer
  Give the position the right image path
- public String getImage(); accessor
  Get the image path of the position

Κλάσεις που κληρονομούν την BoardPosition

- MessagePos
- Sweepstakes
- YardSale
- DealPos
- PayDay
- Buyer
- StartPosition
- RadiContest
- Lottery
- FamilyKazino

Οι εξής κλάσεις RadioContest, Lottery, FamilyKazino, Sweepstakes έχουν το δικό τους attribute

- private int Prize; // the Prize the player wins

Η κλάση MessagePos έχει ως attribute

- private boolean twocards; // if player draw two or one card

Οι υπόλοιπες κλάσεις DealPos, PayDay, Buyer, StartPosition δεν έχουν attributes

Κάθε κλάση κάνει override την μέθοδο toDo από την Parrent Class (BoardPosition) εκτός τις PayDay & StartPosition

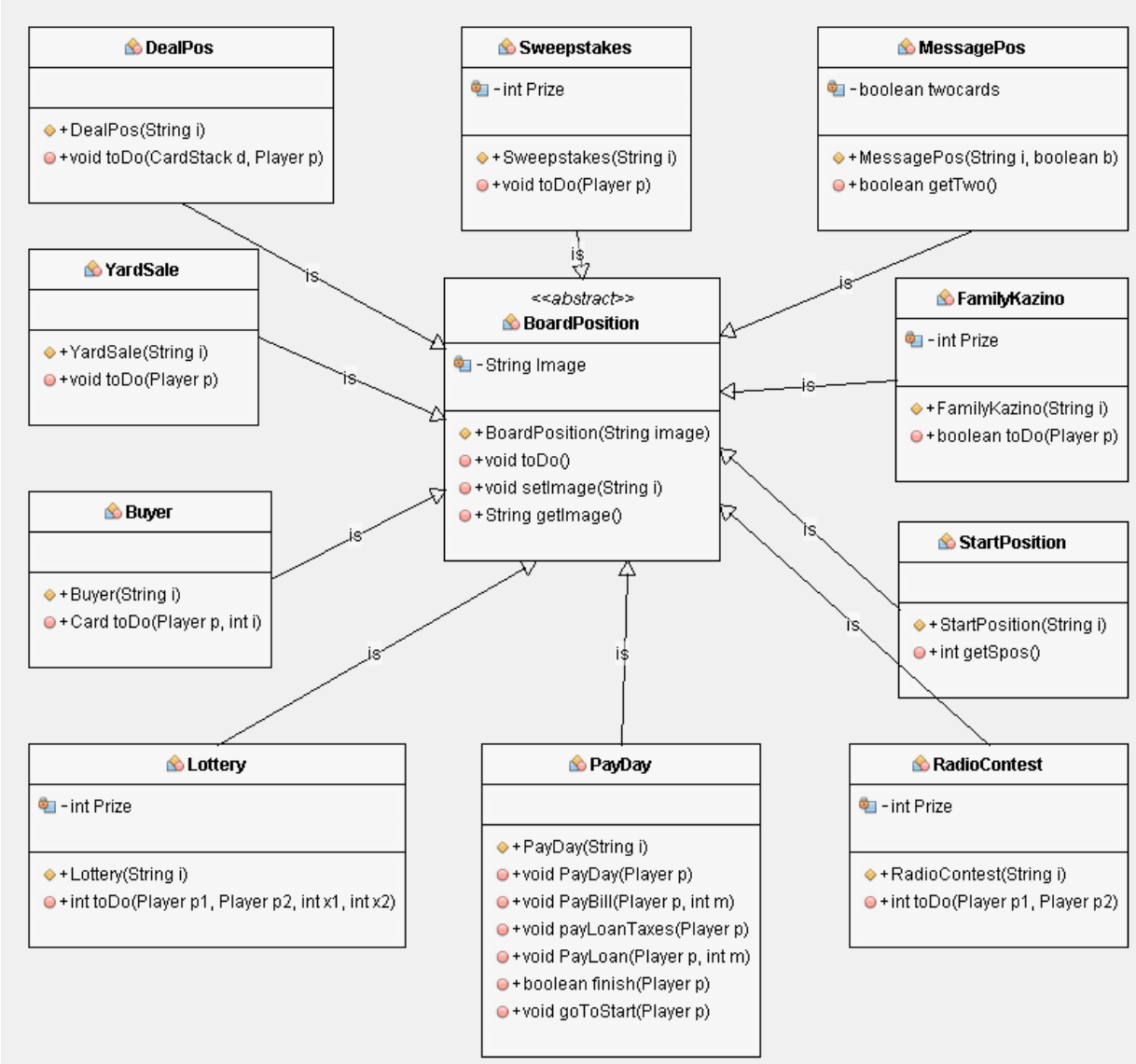
Η StartPosition υλοποιεί την μέθοδο

- public int getSpos(); accessor
  Get the value 0 as the Start position

Η PayDay υλοποιεί τις ακόλουθες μεθόδους

- public void PayDay(Player p); transformer
  Give player that stepped on, money and the opportunity to pay bills and loan , makes him pay loan taxes and stop him if its his final month
- public void PayBill(Player p); transformer
  Let the player pay some bills from his billdept
- public void payLoanTaxes(Player p); transformer
  Player pays 10% of his loan
- public void PayLoan(Player p); transformer
  Let the player to pay a part or all of his loan
- public boolean finish(Player p); accessor
  Return true if player finished the game , false otherwise
- public void goToStart(Player p); transformer
  Set Player's position back to start

Παρακάτω μια αναπαράσταση UML της κλάσης BoardPosition με τις υποκλάσεις τις


DealPos
+DealPos(String i)
+void toDo(CardStack d, Player p)
Sweepstakes
-int Prize
+Sweepstakes(String i)
+void toDo(Player p)
MessagePos
-boolean twocards
+MessagePos(String i, boolean b)
+boolean getTwo()
YardSale
+YardSale(String i)
+void toDo(Player p)
<<abstract>>
BoardPosition
-String Image
+BoardPosition(String image)
+void toDo()
+void setImage(String i)
+String getImage()
FamilyKazino
-int Prize
+FamilyKazino(String i)
+boolean toDo(Player p)
Buyer
+Buyer(String i)
+Card toDo(Player p, int i)
StartPosition
+StartPosition(String i)
+int getSpos()
Lottery
-int Prize
+Lottery(String i)
+int toDo(Player p1, Player p2, int x1, int x2)
PayDay
+PayDay(String i)
+void PayDay(Player p)
+void PayBill(Player p, int m)
+void payLoanTaxes(Player p)
+void PayLoan(Player p, int m)
+boolean finish(Player p)
+void goToStart(Player p)
RadioContest
-int Prize
+RadioContest(String i)
+int toDo(Player p1, Player p2)
is

## Class SundayMatch

Αυτή η κλάση υλοποιεί το σύστημα του στοιχήματος κάθε φορά που ο παίκτης πέσει σε μέρα Κυριακή

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

Attributes:

- private int Prize; // The prize player gets if he wins
- private int Bid; // The bid player sets

Method:

- public void SetBid(int x); transformer
  Set the bid of the player
- public boolean match(); transformer
  Roll the dice and check and return if player won

## Class Dice

Αυτή η κλάση υλοποιεί το ζάρι που χρησιμοποιούν οι πάικτες

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

Attributes:

- private int diceNumber; // Number result after rolling dice
- private boolean thrown; // True if dice is rolled , false otherwise

Method:

- public void RollDice(); transformer
  Take a random number between 1-6 , and set dice thrown true
- public int getDiceNum(); accessor
  Get the number result after rolling dice
- public void setThrown(); transformer
  Set dice thrown to false
- public Boolean getThrown(); accessor
  Get true if the dice is rolled , false otherwise

# Package Controller

## Class Controller

Αυτή η κλάση ελέγχει όλο το παιχνίδι. Είναι υπεύθυνη για την δημιουργία του πίνακα Θέσεων , των παικτών , των καρτών και των ζαριών. Είναι υπεύθυνη για την σύνδεση των view & model. Ουσιαστικά αυτό που κάνει είναι να πέρνει τις επιλογές του χρήστη μέσω των γραφικών και να πραγματοποιεί τις κατάλληλες ενέργειες. Επίσης είναι υπεύθυνη να ενημερώνει πότε τελείωσε το παιχνίδι και τον Νικητή.

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

Attributes:

- private Player player1, player2; // The two players of the game
- private Board board; //The board with the positions(thesis , days)
- private CardStack Dealcards; // Stack with the deal cards
- private CardStack MessageCards; // Stack with the message cards
- private JackPot myJackpot; // Jackpot thesis
- private SundayMatch sm; // Sunday match

Method:

- public Player getP1(); accessor
  Get the first player
- public Player getP2(); accessor
  Get the second player
- public Board getBoard(); accessor
  Get the board of the game
- public CardStack getMstack(); accessor
  Get the Message card's stack
- public CardStack getDstack(); accessor
  Get the Deal card's stack
- public JackPot getJack(); accessor
  Get the JackPot
- public Boolean SMatch(Player p , int b); accessor
  Check player's bid and if he wins returns true
- public void setPlayingMonths(int m); transformer
  Set the desired playing months
- public void BuyDeal(Card c, int am); transformer
  Check player's turn and pay the money to get the card, if player don't have enough money he takes loan
- public void SellDeal(Card c, int i); transformer
  Check whose player is the turn and gives him the money from the card and throw the card to rejected

- public Player FirstPlayer(); accessor
  Set which player will start first
- public Boolean GameFinished(); accessor
  Get if the game is finished
- public Player Winner(); accessor
  Check each player's budget and returns the one with the highest
- public void InitializeGame(); transformer
  Initialize the game (board , players etc)
- public void PlayGame(Player p1,Player p2)
  All the necessary actions to play the game

# Package View

Σε αυτό το πακέτο θα υπάρχει η κλάση GUI που θα υλοποιει τα γραφικά.
Επίσης κ η κλάση test η οποία θα περιέχει την main όπου θα αρχικοποιούνται τα γραφικά κ θα ξεκινάνε.

## Class GUI

Η κλάση αυτή αρχικοποιεί κ υλοποιεί τα γραφικά του παιχνιδιού. Δημιουργεί όλα τα απαραίτητα πάνελ κ κουμπιά που χρειάζονται και προσθέτει τις απαραίτητες εικόνες σ αυτά.

Attributes:

- private JDesktopPane theDesktop; // Basic Panel
- private JTextField infoBox, Money1, Money2, Loan1, Loan2, Dept1, Dept2, Info1, Info2, Info3, Info4; // Text fields for player's information and info box
- private JTextField[] Name; // Array to keep the names(days) of the board positions
- private JPanel Player1, Player2; //Player's panel
- private JPanel[] pall1, pall2; // Player's info panel
- private JButton Dice1, Dice2, Deal1, Deal2, MessageC, DealC; //Buttons
- private ClassLoader cl;
- private JDesktopPane Board; // Board panel
- private URL ImageURL, imageURL; // For image path
- private Image imag; // Image
- private Controller control; // Controller
- private JDesktopPane[] Position; // Array with the positions(days) of the board
- private JLabel d1, d2; // Dice image
- private myDesktopPane pawn1, pawn2; //Player's pawn
- private myDesktopPane[] pos; //Array with the positions

- private boolean draw, get2;
- private int count;

Method:

- void InitializePane(); transformer
  Initialize board , set the panel , players , info box and cards
- void addPFrame(JDesktopPane pane, JPanel Player, JLabel d, JPanel[] pall, JButton Dice, JButton Deal, JTextField Money, JTextField Loan, JTextField Dept, Point p, String Title, Color c) ; transformer
  Create the Player's frame with his informations
- void addPFrame(JDesktopPane pane, JPanel Player, JLabel d, JPanel[] pall, JButton Dice, JButton Deal, JTextField Money, JTextField Loan, JTextField Dept, Point p, String Title, Color c) ; transformer
  Create the Player's frame with his informations
- void addTF(JDesktopPane pane, JTextField I1, JTextField I2, JTextField I3, JTextField I4, Point p) ; transformer
  Create and set the info box
- void addButtonP(JPanel player, JButton button, Point p, Dimension d, String Title) ; transformer
  Buttons for player frame
- void addButton(JDesktopPane pane, JButton button, Point p, Dimension d, String Title, Image img) ; transformer
  Create and set buttons on the basic panel
- void setDiceImg(int x, int p) ; transformer
  Set the dice image according the number it rolled
- void addBoard(JDesktopPane pane, JDesktopPane[] P, Point p, int w, int h) ; transformer
  Create and set the board with the positions on the basic panel
- public void showMailCard(Card c) ; transformer
  According the card drawn , show the necessary informations
- public void showDealCard(Card c) ; transformer
  Show the deal card and its informations
- void setPlayerInfo() ; transformer
  Update both Players information
- void paintPawn(int x) ; transformer
  Paint the pawn of each player according to their position
- void setJackpot() ; transformer
  Update Jackpot informations
- void setInfoText(String s1, String s3, String s4) ; transformer
  Update info box
- void PositionAction(Player p) ; transformer
  According the position of the player on the board , are done the necessary actions

- private int SundayM() ; transformer
  Let the player choose to bid or not
- private void MatchResult(boolean mr) ; transformer
  Show the result of the player's bid , if he won or lost
- private void new_game() ; transformer
  Creates new game