

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστραφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2020-2021**

Logo

Description automatically generated

*Αθανασία Μπουμπουράκη*

*4692*

*8 Δεκεμβρίου 2021*

Pay Daυ

Περιεχόμενα

[1. Εισαγωγή 2](#_Toc93773312)

[2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 3](#_Toc93773313)

[3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller 15](#_Toc93773314)

[4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View 18](#_Toc93773315)

[5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 21](#_Toc93773316)

[6. Λειτουργικότητα 22](#_Toc93773317)

[7. Συμπεράσματα 23](#_Toc93773318)

## Εισαγωγή

Σε αυτό το Project καλούμαστε να υλοποιήσουμε το επιτραπέζιο παιχνίδι “Pay Day”. Το Pay Day είναι ένα επιτραπέζιο παιχνίδι που δημιουργήθηκε αρχικά από τον Parker Brothers το 1975. Εφευρέθηκε από τον Paul J. Gruen από το West Newbury, Μασαχουσέτη, Ηνωμένες Πολιτείες, έναν από τους κορυφαίους σχεδιαστές επιτραπέζιων παιχνιδιών της εποχής και τον αδελφό του Charles C. Μπέιλι. (πηγή:Βικιπέδεια)

Για την ευκολότερη και αποτελεσματικότερη υλοποίηση του παιχνιδιού στη γλώσσα προγραμματισμού Java, χρησιμοποιούμε το προγραμματιστικό μοντέλο MVC ( Model - View – Controller).

Συγκεκριμένα:

❖ Model: πυρήνας παιχνιδιού-περιέχει όλη την πληροφορία κατάστασης

❖ View: υλοποίηση της διεπαφής, δηλαδή το UI (User Interface) ή GUI (Graphical UI)

❖ Controller: συνδετικός κρίκος ανάμεσα σε View και Model

Αυτή η αναφορά περιέχει αναλυτικά τον τρόπο υλοποίησης κάθε ενός από αυτά τα κομμάτια του MVC, καθώς επίσης και τα αντίστοιχα διαγράμματα UML ώστε να γίνει κατανοητός ο σχεδιασμός που επέλεξα για την υλοποίηση του παιχνιδιού.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Σε αυτό το πακέτο περιέχονται οι κλάσεις Player, Board, οι abstract κλάσεις Card, Position και εκείνες που τι κληρονομούν(τύποι καρτών, τύποι θέσεων).

* 1. Class Player:

Αυτή η κλάση αφορά τον παίκτη και περιέχει τα χαρακτηριστικά του(χρήματα, δάνειο, λογαριασμοί, ζάρι, σειρά, κάρτες, αντίπαλος, αν έχει φτάσει στο τέλος).

Τα πεδία της:

1. public ArrayList<Card> cards=new ArrayList<Card>() ;
2. private int money;
3. private int dice;
4. private int loan;
5. private int bills;
6. private boolean turn;
7. private int monthsLeft;
8. private int position=-1;
9. private Player opplayer;
10. boolean hasFinished=false;

Οι μέθοδοι:

1. public Player(); *κοντράκτορας*
2. public int getPosition() ; *επιστρέφει τη θέση του παίκτη*
3. public boolean getTurn() ; *επιστρέφει αν είναι η σειρά του παίκτη ή οχι*
4. public Player getOpPlayer() ; *επιστρέφει τον αντίπαλο του παίκτη*
5. public int getMonths() ; *επιστρέφει πόσοι μήνες απομένουν στον παίκτη*
6. public void setTurn(boolean turn); *θέτει τη σειρά*
7. public void setMonths(int months) ; *αλλάζει τους μηνες που απομένουν*
8. public int getDice() ; *επιστρέφει την τιμή του ζαριού*
9. public int getMoney() ; *επιστρέφει τα χρήματα του παίκτη*
10. public int getLoan() ; *επιστρέφει το δάνειο του παίκτη*
11. public int getBills() ; *επιστρέφει τους λογαριασμούς του*
12. public void gainMoney(int money) ; *προσθέτει χρήματα στον παίκτη*
13. public void setPlayer2(Player opplayer) ; *θέτει τον αντίπαλο*
14. public void payMoney(int money) ; *βάζει τον παίκτη να πληρώσει*
15. public void addLoan(int loan); *ο παίκτης παίρνει δάνειο*
16. public void payLoan(int loan) ; *ξεπληρώνει το δάνειο*
17. public void addBills(int bills); *οι λογαριασμοί του αυξάνονται*
18. public void payBills(int bills); *πληρώνει τους λογαριασμούς*
19. public int rollDice() ; *ρίχνει το ζάρι*
20. public void setPosition(int x) ; *πηγαίνει σε νέα θέση*
21. public void mandatoryLoan(int poso); *πρέπει υποχρεωτικά να πάρει δάνειο*
22. public ArrayList<Card> getCards() ; *οι κάρτες που έχει στην κατοχή του*
23. public void addCards(Card card); *αποκτά καινούρια κάρτα*
24. public void endTurn(); *τελείωσε τη σειρά του*
25. public boolean isHasFinished(); ε*πιστρέφει αν έχει φτάσει στο τέλος*
26. public void setHasFinished(boolean hasFinished); *θέτουμε αν έχει τελειώσει*

Graphical user interface, application

Description automatically generated

* 1. Class Board:

Η κλάση Board αφορά το ταμπλό και περιέχει μια θέση για το jackpot, μια στοίβα για τις mail κάρτες, μια για της deal κάρτες και μια για όσες δεν χρησιμοποιούνται. Είναι υπεύθυνη για το ανακάτεμα των καρτών, την τοποθέτηση όσων δεν χρησιμοποιούνται στην στοίβα trash, την προσθήκη νέων καρτών σε κάθε στοίβα, την αρχικοποίηση των στοιβών αλλά και των καρτών, την αρχικοποίηση του jackpot, τον έλεγχο για το αν κάποια στοίβα είναι κενή ή όχι καθώς και το τι πρέπει να γίνει όταν έχουμε άνοδο κρυπτονομίσματος ή αγώνα Κυριακής.

Τα πεδία της:

1. private int jackpot;
2. private ArrayList<DealCard> dealCards= new ArrayList<DealCard>();
3. private ArrayList<MailCard> mailCards= new ArrayList<MailCard>();
4. private ArrayList<Card> trash= new ArrayList<Card>();

Οι μέθοδοι: *(το τι κάνει η κάθε μια περιγράφεται παραπάνω)*

1. public ArrayList<DealCard> getDealCards() ;
2. public ArrayList<MailCard> getMailCards() ;
3. public void setDealCards(ArrayList<DealCard> dealCards) ;
4. public void setMaillCards(ArrayList<MailCard> mailCards);
5. public void AddDealCardToStack(DealCard C);
6. public void AddMailCardToStack(MailCard C);
7. public boolean isEmptyDeal();
8. public boolean isEmptyMail();
9. public boolean isEmptyTrash();
10. public DealCard TakeDealCard();
11. public MailCard TakeMailCard();
12. public void shuffleMailStack();
13. public void shuffleDealStack();
14. public void throwCard(Card C);
15. public void fillStacks(Board A, Board B) ;
16. public boolean sundayMatch();
17. public boolean crypto() ;
18. public int getJackpot();
19. public void setJackpot(int jack) ;
20. public void addtoJackpot(int jack);

Graphical user interface, application

Description automatically generated

* 1. Class Position:

Η κλάση Position αφορά την θέση στην οποία μπορεί να βρίσκεται ο παίκτης.

Είναι abstract καθώς οι θέσεις μπορεί να είναι είτε για να τραβηχθεί κάρτα είτε για να γίνει ρίψη ζαριού(από τον ένα ή και από τους δύο παίκτες) είτε επειδή είναι το τέλος ή η αρχή του μήνα.

Τα κοινά χαρακτηριστικά όλων των περιπτώσεων είναι όπως φαίνεται και στα πεδία της ο αριθμός πάνω στο ταμπλό, η μέρα και η εικόνα.

Τα πεδία της:

1. private int numOnBoard;
2. private String image;
3. private Days day;

Οι μέθοδοι:

1. Position(int num,String Image);
2. enum Days {

Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

}

1. public int getNum() ;
2. public Days getDay() ;
3. public String getImage() ;
4. public void setNum(int num) ;
5. public void setDay(Days day) ;
6. public boolean isSunday() ; *έλεγχος για τον αν είναι Κυριακή*
7. public boolean isThursday() ; έ*λεγχος για το αν είναι Πέμπτη*
   * 1. Class PayDay:

Αυτή η κλάση κληρονομεί την Position με numOnBoard=31, καθώς είναι η τελευταία μέρα πάνω στο ταμπλό.

Οι μέθοδοι:

1. PayDay(String Image)
2. **public** **void** performAction(Player player) ; *τι θα κάνει όταν φτάσει σε αυτή τη θέση ο παίκτης*

* + 1. Class Start:

Αυτή η κλάση κληρονομεί την Position με numOnBoard=0, καθώς είναι η αρχή του παιχνιδιού πάνω στο ταμπλό.

Οι μέθοδος:

1. Start(String Image) ;
   * 1. Class CardPosition:

Η abstract κλάση CardPosition κληρονομεί την Position και σημαίνει ότι εκεί ο παίκτης πρέπει να τραβήξει μια κάρτα(mail ή deal).

Οι μέθοδοι:

1. CardPosition(int num, String Image);
2. public abstract void getCards(Card C,Player player); *ο παίκτης τραβάει κάρτα αναγκαστικά*
   * 1. Class DealPosition:

Εδώ ο παίκτης πρέπει να τραβήξει μια deal card, γιαυτό κληρονομεί την CardPosition.

Οι μέθοδοι:

1. DealPosition(int num, String Image) ;
2. public void getCards(Card C,Player player) ;
   * 1. Class MailPosition:

Εδώ ο παίκτης πρέπει να τραβήξει μια ή δυο mail card(s), γιαυτό κληρονομεί την CardPosition και έχει ένα επιπλέον πεδίο numOfCards για να ξέρουμε πόσες πρέπει να τραβήξει.

Τα πεδία της:

1. private boolean numOfCards;

Οι μέθοδοι:

1. MailPosition(int num, String Image, boolean number;
2. public void getCards(Card C,Player player,boolean number) ; *η παράμετρος number μας δείχνει πόσες κάρτες θα τραβήξει ο παίκτης*
3. public boolean getNumOfCards() ;
4. public void setNumOfCards(boolean numOfCards) ;
5. public void getCards(Card C, Player player) ;
   * 1. Class DicePosition:

Η abstract αυτή κλάση κληρονομεί την Position και ξέρουμε ότι εδώ ο παίκτης θα χρειαστεί να ρίξει ένα ζάρι για χρήματα(εξ’ού και τα πεδία money και diceNum).

Τα πεδία της:

* 1. private int money;
  2. private int diceNum;

Οι μέθοδοι:

1. DicePosition(int num, String Image) ;
2. public int getMonD();
3. public int getDiceNum() ;
4. public abstract void performAction(Player p, int diceNumber); *εξαρτάται από την θέση που βρίσκεται ο παίκτης*
   * 1. Classes OnePlayer,TwoPlayer:

Αυτές οι δύο κλάσεις κληρονομούν την DicePosition και μας ενημερώνουν για τον αν θα χρειαστεί να ρίξει ο ένας μόνο ή και οι δυο παίκτες ζάρι.

Οι μέθοδοι: (όσες κληρονομούν ίδιες εκτός από τους constructors)

* 1. OnePlayer(**int** num, String Image);
  2. TwoPlayer(**int** num, String Image);
  3. **public** **void** performAction(Player p, **int** diceNumber);
     1. Class Buyer:

Μια ακόμη κλάση που κληρονομεί την Position.Εδώ ο παίκτης μπορεί να πουλήσει μια από τις κάρτες του.

Οι μέθοδοι:

1. Buyer(int num, String Image) ;
2. public void sell(Player player) ; *ο παίκτης επιλέγει ποια κάρτα θα πουλήσει αν έχει κάρτες*
   * 1. Classes YardSale,Sweepstakes,Casino:

Αυτές οι κλάσεις αφορούν θέσεις όπου μόνο ένας παίκτης χρειάζεται να ρίξει ζάρι και γιαυτό όλες κληρονομούν την κλάση OnePlayer.

Οι μέθοδοι: (όσες κληρονομούν ίδιες σε τύπο και όνομα εκτός από τους constructors):

1. Casino(int num, String Image) ;
2. YardSale(int num, String Image) ;
3. Sweepstakes(int num, String Image) ;
4. public void performAction(Player player, Board board) ;*αυτή η συνάρτηση αλλάζει ανάλογα με τη θέση*
   * 1. Classes Radio,Lottery:

Αυτές οι κλάσεις αφορούν θέσεις όπου και οι δυο παίκτες χρειάζεται να ρίξουν ζάρι και γιαυτό όλες κληρονομούν την κλάση TwoPlayer.

Οι μέθοδοι: (όσες κληρονομούν ίδιες σε τύπο και όνομα εκτός από τους constructors)

* 1. Radio(**int** num, String Image) ;
  2. Lottery(**int** num, String Image) ;
  3. **public** **void** performAction(Player p, **int** diceNumber); *αυτή η συνάρτηση αλλάζει ανάλογα με τη θέση*

Graphical user interface, application, Word

Description automatically generated

* 1. Class Card:

Η abstract κλάση Card αφορά τις κάρτες του παιχνιδιού και τα πεδία που θα χρειαστούν οι mail και deal κάρτες είναι το image για την εικόνα και το message για το μήνυμα που θα εμφανίζει.

Τα πεδία της:

1. private String image;
2. private String message;

Οι μέθοδοι:

1. Card(String image,String message);
2. public String getImage() ;
3. public String getMessage() ;
4. public abstract void playerChooses(Player player);
   * 1. Class DealCard:

Αυτή η κλάση κληρονομεί την Card και την εμπλουτίζει με sellCost και buyCost καθώς αφορά της deal κάρτες όπως υποδηλώνει και το ονομά της.

Τα πεδία της:

* 1. int buyCost;
  2. int sellCost;

Οι μέθοδοι:

1. DealCard(String image, String message, int buy, int sell) ;
2. public int getBuycost() ;
3. public int getSellcost() ;
4. public void playerChooses(Player player) ; *δίνεται η δυνατότητα στον παίκτη να αγοράσει μια κάρτα η οποία θα προστεθεί στη συλλογή του και την οποία θα μπορεί να την πουλήσει μετέπειτα*
   * 1. Class MailCard:

Αυτή η κλάση κληρονομεί την Card και την εμπλουτίζει με money(ποσό που ο παίκτης είτε θα κερδίσει είτε θα πληρώσει) καθώς αφορά της deal κάρτες όπως υποδηλώνει και το ονομά της.

Το πεδίο της:

1. int money;

Οι μέθοδοι:

* 1. MailCard(String image, String message,int money) ;
  2. public int getMoney() ;
  3. public void playerChooses(Player player); *διαφοροποιείται ανάλογα το είδος της κάρτας*
     1. Classes GetFromNeighbor, MoveToBS, Ad, PayNeighbor, Bill, Charity:

Αυτές οι κλάσεις αφορούν όλες τις κάρτες τύπου mail οπότε και κληρονομούν την κλάση MailCard.

Οι μέθοδοι: (όσες κληρονομούν ίδιες σε τύπο και όνομα εκτός από τους constructors)

1. Bill(String image, String message, **int** money) ;
2. Ad(String image, String message, **int** money) ;
3. PayNeighbor(String image, String message, **int** money) ;
4. Charity(String image, String message, **int** money) ;
5. MoveToBS(String image, String message, **int** money) ;
6. GetFromNeighbor(String image, String message, **int** money) ;
7. **public** **void** playerChooses(Player player) ; *η κάθε κλάση υλοποιέι αυτή την μέθοδο όπως ζητείται από την εκφώνηση και κάθε φορά είτε θα αφαιρεί είτε θα προσθλετει χρληματα στον παίκτη- ανάλογα το είδος της κάρτας*

Graphical user interface, application, Word

Description automatically generated

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Class Controller:

Σκοπός αυτής της κλάσης είναι έχοντας πρόσβαση στα δεδομένα που συλλέγει το πακέτο model, να ενημερώνει ανάλογα το UI(View) και αντίστοιχα κάθε πληροφορία που προέρχεται από την διεπαφή να περνάει στο model. Πιο αναλυτικά, είναι υπεύθυνη για την αρχικοποίηση του παιχνιδιού, για την ομαλή διεξαγωγή του, την σειρά που θα παίξουν οι παίκτες, την ορθότητα των κινήσεων, τον τερματισμό του παιχνιδιού και την ανάδειξη του νικητή.(όπως ζητείται και στην εκφώνηση)

Τα πεδία αυτής της κλάσης:

1. private String winner ;
2. Board board;
3. View view;
4. Player player1,player2;
5. private ClassLoader cldr;

Οι μέθοδοι:

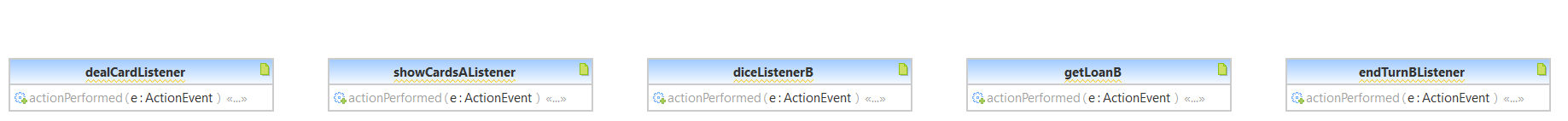
1. public Controller() ;
2. public void initializeGame() ;*αρχικοποιέι το παιχνίδι*
3. public void boardInitializer() ;*αρχικοποιέι το ταμπλο(θέσεις,μέρες κλπ)*
4. public void randomPos(Position[] calendar) ;*βάζει τυχαία κάθε θέση πάνω στο ταμπλό*
5. public void cardInitilizer() ;*αρχικοποιεί τις κάρτες*
6. public void setFirstPlayer() ;*αποφασίζει ποιος θα παίξει πρώτος*
7. public void setMonths() ;*αρχικοποιεί πόσους μήνες θα διαρκέσει το παιχνίδι*
8. public void newMonth() ;*όταν ένας παίκτης πηγαίνει σε καινούριο μήνα*
9. public void movePlayer(ActionEvent e) ;*μετακίνηση πιονιού παίκτη*
10. private void positionActions(Player player, int position) ;*κλήση μεθόδων ανάλογα με τη θέση που βρίσκεται ο παίκτης*
11. public void mailCardAction(Player player, MailCard C) ;*τι θα γίνεται όταν ο παίκτης έχει mail κάρτα*
12. public boolean checkIfGameFinished() ;*έλγχος για το αν ολοκληρώθηκε το παιχνίδι*
13. public void finale() ;*όταν τελειώνει το παιχνίδι*
14. public String[][] readFile(String path, String type) ; *μέθοδος που θα διβάζει ένα αρχείο*

Κλάσεις για τα κουμπιά πάνω στο πάνελ κάθε παίκτη:

1. private class cardListener implements ActionListener ; *θα επιλέγει κάρτα*
2. private class diceListener implements ActionListener ; *θα ρίχνει το ζάρι*
3. private class showCardsListener implements ActionListener ; *θα βλέπει ποιες κάρτες έχει στη συλλογή του*
4. private class endTurnListener implements ActionListener ; *θα ολοκληρώνει τη σειρά του*
5. private class getLoan implements ActionListener ; *θα επιλέγει ποσό για δάνειο*

Graphical user interface, text, application

Description automatically generated



Graphical user interface, application

Description automatically generated

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Class View:

Η κλάση View αφορά την γραφική διεπαφή με την οποία αλληλοεπιδρούν οι τελικοί χρήστες του παιχνιδιού. Το ταμπλό θα είναι μορφής JLayeredPaneExtension και κάθε θέση στο ταμπλό θα είναι ένα JLabel, τα κουμπιά κάθε παίκτη JButton, τα panel των παικτών θα είναι JLabel όπως και τα ζάρια, το jackpot, οι στοίβες με τις κάρτες και η θέση payday. Κάθε κείμενο που θα χρειαστεί να βρίσκεται πάνω στο UI θα είναι JTextField και τα πιόνια των παικτών θα είναι JLayeredPane.Το infobox θα είναι JTextArea όπως προτείνεται και στο παράδειγμα.

Τα πεδία αυτής:

1. private JLabel[] positions;
2. private JLayeredPaneExtension basic\_panel;
3. private JLayeredPane player1;
4. private JLayeredPane player2;
5. private ClassLoader cldr;
6. private JButton endTurn1,rollDice1,getLoan1,showCards1;
7. private JButton endTurn2,rollDice2,getLoan2,showCards2;
8. private JTextField money1, bills1, loan1, player1Name;
9. private JTextField money2, bills2, loan2,player2Name;
10. private JTextField jackpot, turn, infoBoxTitle, monthsLeft, message;
11. JLabel pawn1,pawn2,dice1,dice2,mailStack,dealStack,jackpotPanel,payday;
12. JTextArea infobox;

Οι μέθοδοι:

1. public View() ;
2. public void updatePawn(int position,int player) ;*ενημερώνει τη θέση ενός πιονιού*
3. public void positionsOnBoard() ;*θέτει τις θέσεις πανω στο ταμπλό*
4. public void initComponents(ArrayList<MailCard> mailCards,ArrayList<DealCard> dealCards) ; *αρχικοποιεί τις κάρτες*
5. public void showMailCard(int i,String mailCards[][]) ;*εμφανίζει μια mailcard*
6. public void showDealCard(int i,String dealCards[][]) ;*εμφανίζει μια dealcard*
7. public void cryptoDialog() ;*το popup για την άνοδο του κρυπτονομίσματος*
8. public void sundayMatchDialog() ; *το popup για τον αγώνα Κυριακής*
9. public String chooseMonthDialog() ;*στην αρχή του παιχνιδιού για επιλογή μηνών*
10. public void start() ;*ξεκινάει το παιχνίδι*
11. public void updatePlayerBox(Player player) ;*ενημερώνει το πανελ ενός παίκτη*
12. public void updateJackpot(int money);*ενημερώνει το jackpot*
13. public void updateInfoBox(int monthsLeft,String turn,String message) ;*ενημερώνει το infobox*
14. public void createDices() ;*δημιουργία ζαριών*
15. public void updateDice(int number,String turn);*ενημέρωση ζαριών(για όταν γινεται roll)*
16. public String loanDialog();*όταν πρεπει ο παίκτης να πάρει δάνειο*
17. public String lotteryDialog(Player player);*κατά την λοτταρία οι παίκτες επιλέγουν αριθμό*
18. public String payDayDialog();*ο παίκτης επιλέγει αν θα πληρώσει και πόσο από το δάνειο*
19. public void showMyCards(ArrayList<Card> mycards) ; *όταν ο παίκτης θέλει να δει τις κάρτες του*

Graphical user interface

Description automatically generated with medium confidence

## Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Η κλάση View έρχεται και δίνει μορφή στις κλάσεις Board,Card,Position και Player από το πακέτο model. Με άλλα λόγια, ζωγραφίζει το ταμπλό, τα πιόνια, τις κάρτες, κάθε κουτί με πληροφορία και γενικότερα δίνει ζωή στο παιχνίδι. Με τη βοήθεια της κλάσης controller το view αποκτά επικοινωνία με το model και πλέον όταν ο παίκτης αλληλεπιδρά με το GUI, το model ενημερώνεται άμεσα χάρη στο controller. Κάθε κουμπί πάνω στο ταμπλό κάνει trigger ορισμένες ενέργειες που το controller είναι υπεύθυνο να εκτελεστούν ομαλά και σωστά, ενημερώνωντας παράλληλα το model.

Παρακάτω φαίνεται το πλήρες διάγραμμα UML από την δική μου σχεδίαση του παιχνιδιού:Diagram

Description automatically generated with medium confidence

## Λειτουργικότητα

Κατάφερα να υλοποιήσω κάθε ερώτημα της εκφώνησης πέρα από τα μπόνους ερωτήματα.

Στην αρχή ο παίκτης πρέπει να επιλέξει πόσους μήνες θέλει να κρατήσει το παιχνίδι.

Έπειτα το infobox ενημερώνεται αυτόματα για το ποιος παίκτης θα ξεκινήσει(τυχαία επιλογή) και τα κουμπιά του αντιπάλου απενεργοποιούνται αυτόματα.

Κάθε παίκτης ρίχνει το ζάρι και πρέπει να κάνει όσα του ζητούνται, διαφορετικά δεν μπορεί να τελειώσει τη σειρά του (end turn).

Στις θέσεις που πρέπει να ξαναρίξει το ζάρι ο παίκτης σταματάει και μέχρι να ξαναρίξει δεν μπορεί να συνεχίσει.

Στις θέσεις που πρέπει και οι δυο παίκτες να ρίξουν το ζάρι, πρώτα το παιχνίδι περιμένει ζαριά από αυτόν που έχει σειρά και έπειτα από τον αντίπαλο. Αν η ζαριά είναι ίδια, ο αντίπαλος πρέπει να ξαναρίξει.

Στις θέσεις που πρέπει να τραβήξει κάρτα, ενεργοποιείται η κατάλληλη στοίβα με κάρτες και το παιχνίδι περιμένει από τον παίκτη να τραβήξει.

Στις θέσεις που πρέπει να τραβήξει δύο κάρτες το παιχνίδι περιμένει αντίστοιχα από τον παίκτη να κάνει όσα πρέπει. Αν η πρώτη κάρτα είναι μετακίνηση στην πλησιέστερη θέση αγοραστή, τότε ο παίκτης πρέπει να ξανατραβήξει και έπειτα μετακινείται αυτόματα. Αν και οι δυο είναι για μετακίνηση, ο παίκτης μετακινείται μια φορά.

Όταν κάποιος παίκτης φτάσει στο payday και έχει ακόμα και άλλους μήνες να παίξει, τότε αυτόματα πηγαίνει στην αρχή, αφού όμως πρώτα του ζητηθεί να πληρώσει μέρος του δανείου ανάλογα ρυθμιζόμενο με τα χρήματά του. Αν ήταν ο τελευταίος του μήνας, τότε παραμένει εκεί, ενημερώνεται το πάνελ του, και απενεργοποιούνται τα κουμπιά του, εκτός αν χρειαστεί να συμμετάσχει σε κάποιον άλλο διαγωνισμό.

Όταν και οι δύο παίκτες έχουν τελειώσει εμφανίζεται παράθυρο που ανακοινώνει τον νικητή και δίνει την επιλογή να ξαναπαίξουν ή να τερματίσουν.

## Συμπεράσματα

Σε γενικές γραμμές, η μοντελοποίηση του παιχνιδιού κατά την Α φάση, χρειάστηκε μεθοδικότητα και διορατικότητα καθώς κληθήκαμε να προβλέψουμε τι θα μας φανεί χρήσιμο στην Β φάση του project. Θα έχει αρκετό ενδιαφέρον η παρατήρηση των αλλαγών που θα χρειαστεί να πραγματοποιηθούν στην Β φάση, όπου και τότε θα φανεί μέχρι ποιο βαθμό ήταν σωστή η κατανόηση των ζητούμενων.

Κατά την Β φάση, παρατήρησα μεγάλη δυσκολία στο γραφικό κομμάτι και την μετακίνηση του πιονιού, μέχρι που βρήκα plug ins που μου έκαναν τη ζωή ομολογουμένως πιο εύκολη 😊 . Επίσης συνάντησα δυσκολία στη θέση Radio όπου και οι δυο παίκτες πρέπει να ρίξουν το ζάρι. Κατά τα άλλα, η αρχική σχεδίαση μου στην Α φάση φάνηκε επαρκής καθώς χρειάστηκε να κάνω πολύ μικρές αλλαγές.