Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

PROJECT : PAY DAY

Εισαγωγή

Παπαγερίδης Βασίλειος

ΑΜ : 4710

11/12/2021

Περιεχόμενα

1. Εισαγωγή

Εδώ θα περιγράψετε σε γενικές γραμμές ποιο μοντέλο χρησιμοποιήσατε για την εργασία σας (MVC) και θα αναφέρετε πολύ συνοπτικά τι περιέχουν οι υπόλοιπες ενότητες της αναφοράς.

Η εργασία στηρίζεται πάνω στο μοντέλο Model View Controller (MVC). Σύμφωνα με αυτό, η εφαρμογή αποτελείται από 3 διασυνδεδεμένα τμήματα :

1. Το Model που εμπεριέχει τα δεδομένα (data) που χρησιμοποιούνται από το πρόγραμμα.
2. Το View, το οποίο εμπεριέχει την γραφική διεπαφή του προγράμματος.
3. Τον Controller, ο οποίος δέχεται εισόδους και διαχειρίζεται και ενημερώνει κατάλληλα το View και το Model αντίστοιχα.

Συνεπώς, ο Controller λειτουργεί με τέτοιο τρόπο έτσι ώστε να υπάρχει σύνδεση μεταξύ των Model και View.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακέτο Model αποτελείται από αρκετές μικρές κλάσεις, οι οποίες εμπεριέχουν τα βασικά δεδομένα του παιχνιδιού και υλοποιούν αρκετές λειτουργίες. Αυτά τα στοιχεία συνδεδεμένα με σωστό τρόπο με το πακέτο Controller αποτελούν τον πυρήνα του παιχνιδιού σε επίπεδο μνήμης.

2.1 Player

Η κλάση Player χρησιμοποιείται για την δημιουργία των παικτών του παιχνιδιού. Η κλάση αυτή περιέχει τα βασικά χαρακτηριστικά κάθε παίκτη, τα οποία είναι τα 3 είδη χρημάτων που έχει (λεφτά, λογαριασμοί και δάνεια), το όνομα, οι μήνες που βρίσκεται στο παιχνίδι, ο αντίπαλος του, το ζάρι του, η θέση στην οποία βρίσκεται, καθώς και οι καταστάσεις που βρίσκεται κατά το παιχνίδι, δηλαδή αν είναι η σειρά του, αν έχει ολοκληρώσει την σειρά του ή αναμένεται κάποια ενέργεια από αυτόν.

Πιο συγκεκριμένα :

2.1.1 Attributes:

a) private int balance
Περιγραφή : Τα χρήματα του παίκτη

b) private int bills
Περιγραφή : Οι λογαριασμοί του παίκτη

c) private int loans
Περιγραφή : Τα δάνεια του παίκτη

d) private int monthsCounter
Περιγραφή : Οι μήνες που βρίσκεται ο παίκτης στο παιχνίδι

e) private ArrayList<DealCard> playerCards
Περιγραφή : Οι κάρτες συμφωνίας που έχει ο παίκτης στην κατοχή του

f) private Dice playerDice
Περιγραφή : Το ζάρι του παίκτη

g) private Player opponent
Περιγραφή : Ο αντίπαλος του παίκτη. Στην περίπτωση μας ο αντίπαλος του Player 1 είναι ο Player 2 και το αντίστροφο

h) private boolean playerTurn
Περιγραφή : Μεταβλητή που δείχνει αν είναι η σειρά του παίκτη (true) ή όχι (false)

i) private boolean finish
Περιγραφή : Μεταβλητή που δείχνει αν έχει τελειώσει την σειρά του ο παίκτης (true) ή όχι (false)

j) private boolean pending
Περιγραφή : Μεταβλητή που δείχνει αν αναμένεται κίνηση από τον παίκτη στη σειρά του (true) ή όχι (false)

k) private int position
Περιγραφή : Η θέση του παίκτη επάνω στο ταμπλό

Οι μέθοδοι που δημιουργήσαμε σε αυτή την κλάση θα φανούν πολύ χρήσιμες στην αξιοποίηση των παραπάνω δεδομένων και έχουν σκοπό να διευκολύνουν τη διαχείριση των παικτών. Συγκεκριμένα :

2.1.2 Methods

a) public void setBalance(int balance)
Περιγραφή : Η setBalance δέχεται ως όρισμα έναν ακέραιο, το οποίο είναι ένα χρηματικό ποσό και στη συνέχεια προσθέτει αυτό το ποσό στο υπόλοιπο του

παίκτη.

b) public int getBalance()
Περιγραφή : Η getBalance επιστρέφει τα χρήματα που έχει ο παίκτης στην κατοχή του.

c) public void setLoans(int loans)
Περιγραφή : Η setLoans, δέχεται ως όρισμα έναν ακέραιο, το οποίο είναι ένα χρηματικό ποσό και στη συνέχεια προσθέτει αυτό το ποσό στο υπόλοιπο των δανείων του παίκτη.

d) public int getLoans()
Περιγραφή : Η getLoans επιστρέφει τα δάνεια που συνολικά έχει πάρει ο παίκτης.

e) public void setBills(int bills)
Περιγραφή : Η setBills δέχεται ως όρισμα έναν ακέραιο, το οποίο είναι ένα χρηματικό ποσό και στη συνέχεια προσθέτει αυτό το ποσό στο υπόλοιπο των λογαριασμών του παίκτη.

f) public int getBills()
Περιγραφή : Η getBills επιστρέφει το σύνολο των λογαριασμών που πρέπει ο παίκτης να πληρώσει.

g) public String getName()
Περιγραφή : Η getName επιστρέφει το όνομα του παίκτη (π.χ. Player 1).

h) public void setMonthsCounter(int newMonths)
Περιγραφή : Η setMonthsCounter είναι μία μέθοδος, η οποία δέχεται ως όρισμα τους μήνες που θα διαρκέσει το παιχνίδι και υπολογίζει κάθε φορά πόσοι μήνες απομένουν για την λήξη του παιχνιδιού.

i) public int getMonthsCounter()
Περιγραφή : Η getMonthsCounter επιστρέφει κάθε φορά τους μήνες του παιχνιδιού που απομένουν μέχρι την λήξη του.

j) public void setOpponent(Player opponent)
Περιγραφή : Η setOpponent παίρνει ως όρισμα έναν παίκτη και θέτει αυτόν τον παίκτη ως αντίπαλο του παίκτη που κάλεσε την μέθοδο.

k) public Player getOpponent()
Περιγραφή : Η getOpponent επιστρέφει τον αντίπαλο του εκάστοτε παίκτη.

l) public void setTurn(boolean newTurn)
Περιγραφή : Η setTurn, δέχεται ως όρισμα μία boolean τιμή (true/false), η οποία αντιπροσωπεύει και το αν είναι η σειρά του παίκτη που χρησιμοποιείται η

μέθοδος ή όχι.

m) public boolean getTurn()
Περιγραφή : Η getTurn επιστρέφει true αν είναι η σειρά του παίκτη, διαφορετικά επιστρέφει false.

n) public Dice getDice()
Περιγραφή : Η getDice είναι μία μέθοδος, η οποία μας επιστρέφει το ζάρι του κάθε παίκτη.

o) public void setFinish(boolean finish)
Περιγραφή : Η setFinish δέχεται ως όρισμα μία boolean τιμή (true/false), η οποία δηλώνει για το εάν ο παίκτης έχει τελειώσει (true) ή όχι (false) τη σειρά του.

p) public boolean getFinish()
Περιγραφή : Η getFinish επιστρέφει true αν ο παίκτης έχει τελειώσει τη σειρά του ή false αν δεν έχει τελειώσει τη σειρά του.

q) public void setPending(boolean pending)
Περιγραφή : Η setPending δέχεται ως όρισμα μία boolean τιμή (true/false), η οποία δηλώνει εάν περιμένουμε ο παίκτης να κάνει κάποια ενέργεια στη σειρά του (true) ή όχι (false).

r) public boolean getPending()
Περιγραφή : Η getPending επιστρέφει true αν ο παίκτης αναμένεται να κάνει κάποια ενέργεια στη σειρά του ή false εάν δεν αναμένεται κάποια ενέργεια από αυτόν.

s) public boolean noDealCards()
Περιγραφή : Η noDealCards επιστρέφει true αν ο παίκτης έχει στην κατοχή του κάρτες «Συμφωνίας» ή false αν δεν έχει στην κατοχή του κάρτες «Συμφωνίας».

t) public void takeCard(DealCard newCard)
Περιγραφή : Η takeCard δέχεται ως όρισμα μία κάρτα συμφωνίας (DealCard), η οποία προστίθεται στην λίστα των καρτών που έχει ο παίκτης στην κατοχή του.

u) public DealCard sellCard()
Περιγραφή : Η μέθοδος αυτή στην ουσία αφαιρεί αυτή την κάρτα από την λίστα καρτών του παίκτη και επιστρέφει την κάρτα που αφαιρέθηκε.

v) public void setPosition(int newPosition)
Περιγραφή : Η setPosition δέχεται ως όρισμα έναν ακέραιο, όπου στην ουσία είναι ο αριθμός μίας θέσης επάνω στο ταμπλό και θέτει ως θέση του παίκτη την τιμή του ορίσματος που δέχθηκε.

w) public int getPosition()
Περιγραφή : Η getPosition επιστρέφει τον αριθμό της θέσης, της οποίας ο παίκτης βρίσκεται πάνω στο ταμπλό.

2.2 Dice

Η κλάση Dice χρησιμοποιείται για την δημιουργία του ζαριού του παιχνιδιού. Συγκεκριμένα περιέχει τα βασικά γνωρίσματα ενός εξάεδρου ζαριού που είναι ο αριθμός (1-6), καθώς και το αν το ζάρι βρίσκεται σε κατάσταση «ρίψης» από έναν παίκτη. Πιο συγκεκριμένα :

2.2.1 Attributes :

a) private int number
Περιγραφή : ο αριθμός του ζαριού

b) private boolean roll
Περιγραφή : τιμή αληθείας για το αν το ζάρι ρίχνεται ή όχι

Οι μέθοδοι αυτής της κλάσης έχουν ως σκοπό να κάνουν το ζάρι του παιχνιδιού πλήρως λειτουργικό και να λειτουργεί ομαλά για κάθε παίκτη του παιχνιδιού.

2.2.2 Methods :

a) public void setNumber(int newNumber)
Περιγραφή : Η setNumber δέχεται ως όρισμα έναν ακέραιο αριθμό, από το 1 εώς το 6 και στη συνέχεια θέτει ως τιμή του ζαριού τον αριθμό αυτό.

b) public int getNumber()
Περιγραφή : Η getNumber επιστρέφει τον αριθμό που έχει το ζάρι εκείνη τη στιγμή.

c) public void setRoll(boolean newRoll)
Περιγραφή : Η setRoll δέχεται ως όρισμα μια boolean τιμή (true/false), η οποία δηλώνει αν το ζάρι πραγματοποιεί ρίψη (true) ή όχι (false) και ενημερώνει την κατάσταση του ζαριού.

d) public boolean getRoll()
Περιγραφή : Η getRoll επιστρέφει την κατάσταση του ζαριού, δηλαδή αν πραγματοποιεί ρίψη (true) ή όχι (false).

e) public int rollTheDice()
Περιγραφή : Η rollTheDice επιστρέφει μία τυχαία τιμή από το 1 ως το 6, έτσι ώστε κάθε φορά που γίνεται ρίψη του ζαριού να υπάρχει η τυχαία τιμή της κάθε

ρίψης που θα χρησιμοποιηθεί για να αλλάξει η τιμή του ζαριού μέσω της setNumber.

2.3 Jackpot

Η κλάση Jackpot χρησιμοποιείται για την υλοποίηση του Jackpot που προσφέρει το παιχνίδι στους παίκτες. Περιέχει το βασικό στοιχείο του Jackpot, δηλαδή το χρηματικό ποσό που προσφέρεται και τις μεθόδους που κάνουν λειτουργικό το Jackpot κατά τη διάρκεια του παιχνιδιού. Συγκεκριμένα :

2.3.1 Attributes :

a) private int money
Περιγραφή : Το χρηματικό ποσό του Jackpot

2.3.2 Methods :

a) public void setMoney(int newMoney)
Περιγραφή : Η setMoney δέχεται ως όρισμα έναν ακέραιο, ο οποίος είναι ένα χρηματικό ποσό αυστηρά μεγαλύτερο του μηδενός και το προσθέτει στο υπόλοιπο του Jackpot.

b) public int getMoney()
Περιγραφή : Η getMoney επιστρέφει έναν ακέραιο, ο οποίος είναι το ποσό του Jackpot τη δεδομένη στιγμή.

c) public int jackpotAward()
Περιγραφή : Η jackpotAward επιστρέφει έναν ακέραιο, ο οποίος είναι το ποσό που κερδίζει ο παίκτης από το Jackpot. Έπειτα το ποσό του Jackpot γίνεται ξανά μηδενικό.

d) public String toString()
Περιγραφή : Η toString στη συγκεκριμένη κλάση επιστρέφει τα λεφτά τα οποία εμπεριέχονται στο Jackpot.

Για την καλύτερη οργάνωση αλλά και για δική μας διευκόλυνση δημιουργήσαμε 2 Group (folders) που περιέχουν τα abstract class στα οποία βασίζονται τα υπόλοιπα classes κάθε group. Τα group αυτά είναι οι θέσεις (positions) και οι κάρτες (cards) του παιχνιδιού.

2.4 Position (Group)

Ο λόγος για τον οποίο χωρίσαμε τις κλάσεις κατά αυτό τον τρόπο είναι διότι όλες οι θέσεις έχουν κάποια κοινά γνωρίσματα και εκμεταλλευόμενοι αυτό το γεγονός κάνουμε τον κώδικα μας πιο κατανοητό και καθαρό, καθώς είναι διαιρεμένος σε subclasses. Επίσης, ο κώδικας μας δεν επαναλαμβάνεται για ίδια πράγματα ξανά και ξανά και στηρίζεται πάνω στις ίδιες μεθόδους που μπορούν να διαφοροποιούνται ανάλογα με το είδος της θέσης και της ανάγκες της.

2.4.1 Position

Η abstract class Position έχει ως σκοπό την υλοποίηση της ιδέας των θέσεων επάνω στο ταμπλό του παιχνιδιού. Απαρτίζεται από τα βασικά γνωρίσματα μίας θέσης όπως η αριθμητική θέση της στο ταμπλό, η εικόνα της θέσης και το όνομα, αλλά και από τις μεθόδους που αποσκοπούν στη λειτουργικότητα των θέσεων. Πιο συγκεκριμένα :

2.4.1.1 Attributes :

a) private int index
Περιγραφή : Η αριθμητική θέση της θέσης επάνω στο ταμπλό

b) private String name
Περιγραφή : Το όνομα της θέσης

c) private final String image
Περιγραφή : Η εικόνα της θέσης

2.4.1.2 Methods:

a) public void setIndex(int newIndex)
Περιγραφή : Η setIndex δέχεται ως όρισμα έναν ακέραιο, ο οποίος αποτελεί την αριθμητική τιμή της θέσης στο ταμπλό και στην συνέχεια ενημερώνει την αριθμητική θέση της θέσης.

b) public int getIndex()
Περιγραφή : Η getIndex επιστρέφει την αριθμητική θέση μίας θέσης

c) public void setName(String name)
Περιγραφή : Η setName δέχεται ως όρισμα ένα string, το οποίο χρησιμοποιείται για την ανάθεση του ονόματος της θέσης.

d) public String getName()
Περιγραφή : Η getName επιστρέφει ένα String, το οποίο είναι συγκεκριμένα το όνομα της θέσης.

e) public String getImage()
Περιγραφή : Η getImage επιστρέφει ένα String που είναι συγκεκριμένα το URL

της εικόνας της θέσης.

f) public boolean isThursday()
Περιγραφή : Η isThursday επιστρέφει μία boolean τιμή (true/false), όπου αν η επιστρεφόμενη τιμή είναι true σημαίνει ότι η μέρα σε εκείνη τη θέση του παιχνιδιού είναι ημέρα «Πέμπτη», διαφορετικά επιστρέφει false.

g) public boolean isSunday()
Περιγραφή : Η isSunday επιστρέφει μία boolean τιμή (true/false), όπου αν η επιστρεφόμενη τιμή είναι true σημαίνει ότι η μέρα σε εκείνη τη θέση του παιχνιδιού είναι ημέρα «Κυριακή», διαφορετικά επιστρέφει false.

Όλα τα παρακάτω class του Group "position" κάνουν extend το abstract class Position που περιγράψαμε πιο πάνω.

2.4.2 MailCardPosition

Η κλάση MailCardPosition χρησιμοποιείται για την υλοποίηση της «Θέσης μηνύματος» στο παιχνίδι.

2.4.2.1 Attributes :

a) private final boolean draw
Περιγραφή : boolean μεταβλητή που δηλώνει αν ο παίκτης πρέπει να τραβήξει κάποια κάρτα (true) ή όχι (false).

2.4.2.2 Methods :

a) public boolean drawCard()
Περιγραφή : Η drawCard επιστρέφει στην ουσία τον αριθμό των καρτών που θα τραβήξει ο παίκτης, μέσω της τιμής που θα επιστρέψει η μέθοδος (που είναι true ή false).

2.4.3 DealPosition

Η κλάση DealPosition χρησιμοποιείται για την υλοποίηση της «Θέσης συμφωνίας» στο παιχνίδι. Η κλάση αυτή αποτελείται μόνο από τον Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και

συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.4 Sweeptakes

Η κλάση Sweeptakes χρησιμοποιείται για την υλοποίηση της θέσης «Λαχείο» στο παιχνίδι. Ο Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.4.1 Methods :

a) public void action(Player p, int index)
Περιγραφή : Η action δέχεται ως ορίσματα έναν παίκτη και τον αριθμό που έδειξε το ζάρι. Η μέθοδος αυτή φροντίζει έτσι ώστε ο παίκτης που βρίσκεται σε αυτή τη θέση να πάρει σαν χρηματικό κέρδος τον αριθμό της ζαριάς του πολλαπλασιασμένο επί 1000.

2.4.5 RadioContest

Η κλάση RadioContest χρησιμοποιείται για την υλοποίηση της θέσης «Διαγωνισμός στο Ραδιόφωνο» στο παιχνίδι. Η κλάση αυτή αποτελείται μόνο από τον Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.6 Buyer

Η κλάση Buyer χρησιμοποιείται για την υλοποίηση της «Θέσης Αγοραστή» στο παιχνίδι. Η κλάση αυτή αποτελείται μόνο από τον Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.7 FamilyCasino

Η κλάση FamilyCasino χρησιμοποιείται για την υλοποίηση της θέσης «Βραδιά Οικογενειακού Καζίνο» στο παιχνίδι. Ο Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.7.1 Methods :

a) public boolean action(Player p, Jackpot jackpot)
Περιγραφή : Η action δέχεται ως ορίσματα έναν παίκτη και το jackpot του

παιχνιδιού. Αν η ζαριά του παίκτη είναι ζυγός αριθμός, τότε ο παίκτης κερδίζει 500 ευρώ από το Jackpot και η μέθοδος επιστρέφει true (δηλαδή ότι ο παίκτης κέρδισε), ενώ αν ο αριθμός της ζαριάς του είναι μονός αριθμός ο παίκτης αφήνει 500 ευρώ από το υπόλοιπο του στο Jackpot του παιχνιδιού και η μέθοδος επιστρέφει false.

2.4.8 YardSale

Η κλάση YardSale χρησιμοποιείται για την υλοποίηση της θέσης «Αγορά με Έκπτωση» στο παιχνίδι. Ο Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 30 και συγχρόνως να μην έχει παρθεί αυτό το index από άλλη θέση, και από το URL της εικόνας της θέσης.

2.4.8.1 Methods :

a) public void action(Player p, int index, DealCard C)
Περιγραφή : Η action δέχεται ως ορίσματα έναν παίκτη, τον αριθμό του ζαριού και μια κάρτα συμφωνίας (DealCard). Η μέθοδος αυτή αφαιρεί τον αριθμό του ζαριού πολλαπλασιασμένο επί 100 από το υπόλοιπο του παίκτη και προσθέτει μία κάρτα στη λίστα καρτών που έχει στην κατοχή του.

2.4.9 PayDayPosition

Η κλάση YardSale χρησιμοποιείται για την υλοποίηση της θέσης «Αγορά με Έκπτωση» στο παιχνίδι. Ο Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο είναι ίσο με 31 αφού είναι η τελευταία θέση στο ταμπλό, και από το URL της εικόνας της θέσης.

2.4.9.1 Methods :

a) public void action(Player p, int loanOrPart, int partOfLoan)*
Περιγραφή : Η action δέχεται ως ορίσματα έναν παίκτη, έναν ακέραιο που δηλώνει αν θα πληρώσει το δάνειο ή ένα μέρος του δανείου, καθώς και έναν ακέραιο, ο οποίος πρόκειται για ένα χρηματικό ποσό δανείου που θα πληρώσει ο παίκτης στην PayDay. Αυτό το ποσό μπορεί να είναι όλο το δάνειο ή ένα μέρος του δανείου. Έπειτα αυτή η μέθοδος υλοποιεί και με την βοήθεια άλλων μεθόδων όλες τις ενέργειες που πρέπει να πραγματοποιηθούν για έναν παίκτη στην PayDay.

2.4.10 Start

Η κλάση Start χρησιμοποιείται για την υλοποίηση της θέσης έναρξης στο παιχνίδι. Η κλάση αυτή αποτελείται μόνο από τον Constructor, ο οποίος χρησιμοποιεί τον Constructor του γονέα και θέτει το index της θέσης, το οποίο είναι

ίσο με 0 αφού αποτελεί την αφετηρία του παιχνιδιού, και από το URL της εικόνας της θέσης.

Όπως γίνεται αντιληπτό, σε πολλά από τα subclasses υπάρχει μόνο ο Constructor. Σε αυτή την φάση και στο συγκεκριμένο πακέτο δίνουμε περισσότερο έμφαση στα δεδομένα και συνεπώς για τον σκοπό των πράξεων θα χρησιμοποιήσουμε το Controller που στην ουσία αποτελεί τον «εγκέφαλο» του προγράμματος μας.

2.5 Cards (Group)

Ο λόγος για τον οποίο χωρίσαμε τις κλάσεις κατά αυτό τον τρόπο είναι ο ίδιος για τον οποίο χωρίσαμε και τις κλάσεις στο group "position". Επιπλέον, εδώ δημιουργήσαμε και άλλα subclasses για τα MailCards, καθώς βοηθούν στην ευκολότερη ανάπτυξη αλλά και δομή του προγράμματος για τους λόγους που έχουμε ήδη αναφέρει. Τέλος, στο ίδιο group περιλαμβάνονται και οι στοίβες όλων των κατηγοριών των καρτών.

2.5.1 Card

Η abstract class Card χρησιμοποιείται για την υλοποίηση της ιδέας των διαφόρων ειδών καρτών που υπάρχουν στον παιχνίδι. Αποτελείται από τα κοινά γνωρίσματα και μεθόδους που κληρωνόμουν όλα τα είδη των καρτών. Πιο συγκεκριμένα :

2.5.1.1 Attributes :

a) private final String image
Περιγραφή : Το URL της εικόνας της κάρτας.

b) private final int money
Περιγραφή : Το χρηματικό κόστος της κάρτας.

c) private final String text
Περιγραφή : Το κείμενο της κάρτας.

2.5.1.2 Methods :

a) public String getImage()
Περιγραφή : Η getImage επιστρέφει το URL της εικόνας της κάρτας.

b) public int getMoney()
Περιγραφή : Η getMoney επιστρέφει έναν ακέραιο, ο οποίος είναι το χρηματικό

κόστος της κάρτας.

c) public String getText()
Περιγραφή : Η getText επιστρέφει το μήνυμα της κάρτας.

d) public abstract void action(Player p)
Περιγραφή : Η abstract μέθοδος action δέχεται ως όρισμα έναν παίκτη, πάνω στον οποίο γίνεται μία ενέργεια. Την μέθοδο αυτή κληρωνόμουν και κάνουν Override όλα τα subclasses της Card class.

2.5.2 DealCard

Η κλάση DealCard κάνει extend την abstract class Card και χρησιμοποιείται για την υλοποίηση των καρτών «συμφωνίας» του παιχνιδιού. Πέρα από τα στοιχεία που κληρονομεί, η κλάση αυτή έχει κάποια στοιχεία παραπάνω, καθώς οι κάρτες συμφωνίας έχουν και μία τιμή πώλησης. Πιο αναλυτικά :

2.5.2.1 Attributes :

a) private final int sell
Περιγραφή : Η τιμή πώλησης της κάρτας.

2.5.2.2 Methods :

a) public int getSell()
Περιγραφή : Η getSell επιστρέφει έναν ακέραιο, ο οποίος είναι η τιμή πώλησης της κάρτας.

b) public void action(Player p)
Περιγραφή : Η action σε αυτή την κλάση δέχεται ως όρισμα ένα παίκτη (όπως είδαμε πριν) και η ενέργεια της είναι : Αν ο παίκτης αγοράσει την κάρτα, τότε αυτή προστίθεται στην λίστα καρτών που έχει, διαφορετικά πηγαίνει στην στοίβα από τις κάρτες που έχουν απορριφθεί.

2.5.3 MailCard

Η abstract class MailCard κάνει extend την abstract class Card και χρησιμοποιείται για την υλοποίηση των καρτών «μηνυμάτων» του παιχνιδιού. Η κλάση αυτή είναι ένα abstract class, διότι υπάρχουν διάφορα είδη MailCard, τα οποία μοιράζονται τα ίδια στοιχεία. Έτσι ο κώδικας γίνεται πιο εύχρηστος και πιο κατανοητός.

2.5.3.1 Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη, ο οποίος πραγματοποιεί την ενέργεια της κάρτας. Την μέθοδο αυτή κληρωνόμουν όλες οι κλάσεις που κάνουν extend την MailCard.

2.5.4 PayTheNeighbor

Η κλάση PayTheNeighbor χρησιμοποιείται για την υλοποίηση της κάρτας «Πλήρωσε τον γείτονα» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο), το χρηματικό ποσό της κάρτας (μεγαλύτερο ή ίσο του 0) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα, το χρηματικό ποσό και το μήνυμα της κάρτας.

2.5.4.1 Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση μεταφέρει ένα χρηματικό ποσό από το υπόλοιπο του παίκτη στον αντίπαλο του.

2.5.5 TakeFromNeighbor

Η κλάση TakeFromNeighbor χρησιμοποιείται για την υλοποίηση της κάρτας «Πάρε λεφτά από το γείτονα» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο), το χρηματικό ποσό της κάρτας (μεγαλύτερο ή ίσο του 0) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα, το χρηματικό ποσό και το μήνυμα της κάρτας.

2.5.5.1 Method :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση ο παίκτης παίρνει ένα χρηματικό ποσό από το υπόλοιπο του αντιπάλου του.

2.5.6 Charity

Η κλάση Charity χρησιμοποιείται για την υλοποίηση της κάρτας «Φιλανθρωπία» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο), το χρηματικό κόστος της κάρτας (μεγαλύτερο ή ίσο του 0) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα, το χρηματικό κόστος και το μήνυμα της εικόνας.

2.5.6.1 : Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση αφαιρείται το χρηματικό ποσό που αναγράφεται από το υπόλοιπο του παίκτη και προστίθεται στο υπόλοιπο του Jackpot.

2.5.7 Bill

Η κλάση Bill χρησιμοποιείται για την υλοποίηση της κάρτας «Εξόφληση Λογαριασμού» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο), το χρηματικό κόστος της κάρτας (μεγαλύτερο ή ίσο του 0) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα, το χρηματικό κόστος και το μήνυμα της εικόνας.

2.5.7.1 Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση προσθέτει το πόσο που αναγράφεται στο υπόλοιπο των λογαριασμών του παίκτη, ώσπου να το εξοφλήσει.

2.5.8 MoveToDealBuyer

Η κλάση MoveToDealBuyer χρησιμοποιείται για την υλοποίηση της κάρτας «Μετακίνηση στην πλησιέστερη θέση Συμφωνίας/Αγοραστή» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα και το μήνυμα της εικόνας.

2.5.8.1 Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση μετακινεί τον παίκτη στην πλησιέστερη θέση συμφωνίας ή αγοραστή.

2.5.9 Advertisement

Η κλάση Advertisement χρησιμοποιείται για την υλοποίηση της κάρτας «Διαφήμισης» στο παιχνίδι. Ο Constructor, ο οποίος δέχεται ως ορίσματα το URL της εικόνας της κάρτας (που πρέπει να είναι έγκυρο) και το μήνυμα της κάρτας, στη συνέχεια χρησιμοποιεί τον Constructor του γονέα και θέτει με βάση τα ορίσματα που δέχεται, την εικόνα και το μήνυμα της εικόνας.

2.5.9.1 Methods :

a) public void action(Player p)
Περιγραφή : Η action δέχεται ως όρισμα έναν παίκτη και στη συγκεκριμένη κλάση πουλάει την συγκεκριμένη κάρτα και προσθέτει στο υπόλοιπο του παίκτη το ποσό που αναγράφεται στην κάρτα.

2.5.10 AllCardsStack

Η κλάση AllCardsStack χρησιμοποιείται για την υλοποίηση μίας στοίβας, στην οποία πηγαίνουν όλες οι κάρτες οι οποίες έχουν απορριφθεί κατά τη διάρκεια του παιχνιδιού. Πιο συγκεκριμένα αποτελείται από :

2.5.10.1 Attributes :

a) private ArrayList<Card> rejectedCardStack
Περιγραφή : Η στοίβα απόρριψης

2.5.10.2 Methods :

a) public void push(Card c)
Περιγραφή : Η push δέχεται ως όρισμα μία κάρτα και την προσθέτει στη στοίβα απόρριψης

b) public void popAndSeperate(MailCardsStack mailCards, DealCardsStack dealCards)
Περιγραφή : Η popAndSeperate δέχεται ως ορίσματα τη στοίβα των MailCards και τη στοίβα των DealCards και στην ουσία αφαιρεί μία κάρτα από τη στοίβα απόρριψης και την προσθέτει στην κατάλληλη στοίβα.

c) public Boolean isStackEmpty()
Περιγραφή : Η isStackEmpty επιστρέφει true αν η στοίβα είναι κενή ή false αν η στοίβα περιέχει κάρτες.

2.5.11 DealCardsStack

Η κλάση DealCardsStack χρησιμοποιείται για την υλοποίηση μίας στοίβας που περιέχει τις κάρτες «Συμφωνίας».

2.5.11.1 Attributes :

a) private ArrayList<DealCard> dealCardStack
Περιγραφή : Η στοίβα των καρτών «Συμφωνίας».

2.5.11.2 Methods :

a) public void push(Card c)
Περιγραφή : Η push δέχεται ως όρισμα μία κάρτα και την προσθέτει στη στοίβα καρτών «Συμφωνίας».

b) public *DealCard pop()
Περιγραφή : Η pop αφαιρεί μία κάρτα από την στοίβα καρτών «Συμφωνίας» και επιστρέφει την κάρτα που αφαιρέθηκε.

c) public void suffleCardStack()
Περιγραφή : Η suffleCardStack ανακατεύει τις κάρτες στην στοίβα καρτών «Συμφωνίας».

d) public Boolean isStackEmpty()
Περιγραφή : Η isStackEmpty επιστρέφει true αν η στοίβα είναι κενή ή false αν η στοίβα περιέχει κάρτες.

2.5.12 MailCardsStack

Η κλάση MailCardsStack χρησιμοποιείται για την υλοποίηση μίας στοίβας που περιέχει τις κάρτες «Μηνύματος».

2.5.12.1 Attributes :

a) private ArrayList<MailCard> MailCardStack
Περιγραφή : Η στοίβα των καρτών «Μηνύματος».

2.5.12.2 Methods :

a) public void push(Card c)
Περιγραφή : Η push δέχεται ως όρισμα μία κάρτα και την προσθέτει στη στοίβα καρτών «Μηνύματος».

b) public void pop()
Περιγραφή : Η pop αφαιρεί μία κάρτα από την στοίβα καρτών «Μηνύματος».

c) public void suffleCardStack()
Περιγραφή : Η suffleCardStack ανακατεύει τις κάρτες στην στοίβα καρτών «Μηνύματος».

d) public Boolean isStackEmpty()
Περιγραφή : Η isStackEmpty επιστρέφει true αν η στοίβα είναι κενή ή false αν η στοίβα περιέχει κάρτες.

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Το πακέτο Controller αποτελείται από μία κλάση (Controller), όπου στην ουσία αποτελεί τον εγκέφαλο του παιχνιδιού. Συγκεκριμένα τα δεδομένα που αποθηκεύονται με την χρήση των κλάσεων του Model, συλλέγει αυτά τα δεδομένα, τα επεξεργάζεται κατάλληλα και στη συνέχεια αλληλοεπιδρά με το View ανανεώνοντας κάθε φορά το τελικό αποτέλεσμα του.

3.1 Controller

Η κλάση Controller αποτελεί τον εγκέφαλο του παιχνιδιού. Όλες οι ενέργειες για την αρχικοποίηση του παιχνιδιού πραγματοποιούνται εδώ. Επιπλέον, συλλέγει μέσω των γραφικών τις ενέργειες του κάθε παίκτη και σε συνδυασμό με τα δεδομένα που παρέχονται από το Model πραγματοποιεί τις κατάλληλες ενέργειες για την ομαλή λειτουργία του παιχνιδιού και την ενημέρωση τόσο των δεδομένων, όσο και των γραφικών. Λόγω του πλήθους των ενεργειών που εκτελεί το Controller θα τις περιγράψουμε ως εξής :

3.1.1 Attributes :

a) public Player p1, p2
Περιγραφή : Οι παίκτες του παιχνιδιού.

b) public Position table[]
Περιγραφή : Πίνακας, ο οποίος κρατάει τις θέσεις του ταμπλό.

c) public AllCardsStack RejectedCards
Περιγραφή : Η στοίβα απόρριψης καρτών.

d) public MailCardsStack MailCardsStack
Περιγραφή : Η στοίβα καρτών «Μηνύματος».

e) public DealCardsStack DealCardsStack
Περιγραφή : Η στοίβα καρτών «Συμφωνίας».

f) public boolean start
Περιγραφή : Boolean τιμή που αντιστοιχεί στο αν το παιχνίδι έχει αρχίσει (true) ή όχι (false).

g) public int pointsP1
Περιγραφή : Οι πόντοι του 1ου παίκτη του παιχνιδιού.

h) public int pointsP2
Περιγραφή : Οι πόντοι του 2ου παίκτη του παιχνιδιού.

i) public Jackpot jackpot
Περιγραφή : Το Jackpot του παιχνιδιού.

j) *private ClassLoader cldr
Περιγραφή : Χρησιμοποιείται για να φορτώνει δυναμικά κλάσεις στην εικονική μηχανή της Java (JVM).

3.1.2 Methods :

a) public void initGame()
Περιγραφή : Η initGame πρόκειται για την μέθοδο που αρχικοποιεί το παιχνίδι.

b) public void initTable()
Περιγραφή : Η initTable αρχικοποιεί το ταμπλό του παιχνιδιού και τοποθετεί τις θέσεις επάνω στο ταμπλό.

c) public static void positionSuffle(Position[] pos)
Περιγραφή : Η positionSuffle δέχεται ως όρισμα έναν πίνακα τύπου Position (θέσεων), ο οποίος είναι γεμάτος και αυτό που κάνει είναι να ανακατεύει αυτές τις θέσεις, εκτός της πρώτης θέσης (θέση αφετηρίας) και της τελευταίας (PayDay).

d) public void gameMonths(int months)
Περιγραφή : Η gameMonths δέχεται ως όρισμα έναν ακέραιο, ο οποίος είναι μεγαλύτερος ή ίσος του 1 και μικρότερος ή ίσος του 3 και θέτει τους μήνες που θα διαρκέσει το παιχνίδι.

e) public void initCards()
Περιγραφή : Η initCards αρχικοποιεί όλες τις κάρτες του παιχνιδιού.

f) public void playingFirst(int p1_dice, int p2_dice)
Περιγραφή : Η playingFirst δέχεται ως ορίσματα 2 ακέραιους αριθμούς, οι οποίοι είναι μία ζαριά κάθε παίκτη (διαφορετικού αποτελέσματος) και ανάλογα με τη μεγαλύτερη ζαριά ορίζει τον παίκτη που θα παίξει πρώτος στο παιχνίδι.

g) public void initBalance()
Περιγραφή : Η initBalance αρχικοποιεί το υπόλοιπο και των 2 παικτών.

h) public void jackpot(Player p)
Περιγραφή : Η jackpot δέχεται ως όρισμα έναν παίκτη και πρόκειται να υλοποιεί τις κατάλληλες ενέργειες όταν ένας παίκτης πέφτει στο Jackpot του παιχνιδιού.

i) public boolean isThursday(Player p, int bet)
Περιγραφή : Η isThursday δέχεται ως ορίσματα έναν παίκτη και μία επιλογή πονταρίσματος. Επιστρέφει true αν ο παίκτης κερδίζει σύμφωνα με την επιλογή του ή false αν ο παίκτης δεν κερδίζει.

j) public boolean isSunday(Player p, int bet)
Περιγραφή : Η isSunday δέχεται ως ορίσματα έναν παίκτη και μία επιλογή στοιχηματισμού. Επιστρέφει true αν ο παίκτης κερδίζει το δελτίο του ή false αν ο

παίκτης δεν κερδίζει.

k) public boolean gameFinish()
Περιγραφή : Η gameFinish σηματοδοτεί την λήξη ή όχι του παιχνιδιού. Αν το παιχνίδι έχει τελειώσει επιστρέφει true, αλλιώς επιστρέφει false.

l) public String gameWinner()
Περιγραφή : Η gameWinner υπολογίζει και συγκρίνει τα σκορ των παικτών και με βάση αυτά καθορίζει και επιστρέφει τον νικητή του παιχνιδιού.

m) public void throwAllCards(Player p)
Περιγραφή : Η throwAllCards δέχεται ως όρισμα έναν παίκτη και ενεργεί έτσι ώστε να πετάξει όλες τις κάρτες από τη συλλογή καρτών του.

n) public String[] gameLog()
Περιγραφή : Η gameLog επιστρέφει ένα String Array, το οποίο είναι χρήσιμες πληροφορίες που προβάλλονται στους παίκτες κατά τη διάρκεια του παιχνιδιού.

o) *public String[][] readFile(String path, String type)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να διαβάζει το αρχείο με τις κάρτες του παιχνιδιού (μας δίνεται από τον διδάσκοντα).

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View θα περιέχει την βασική κλάση View που αποτελεί την υλοποίηση της ιδέας των κύριων γραφικών χαρακτηριστικών του ταμπλό, των θέσεων επάνω στο ταμπλό, τα panels των παικτών (καθώς και τα στοιχεία και τα κουμπιά που βρίσκονται μέσα σε αυτά τα panel), το μενού του παιχνιδιού, το Jackpot, τις στοίβες των καρτών, το ζάρι, τα πιόνια των παικτών και τα γραφικά παράθυρα που θα πετάγονται κατά την διάρκεια του παιχνιδιού.

Επιπλέον, επειδή υπάρχουν κάποια παράθυρα που θα πρέπει να σχεδιάσουμε για συγκεκριμένες ενέργειες και καταστάσεις μέσα στο παιχνίδι, έχουμε δημιουργήσει τις κλάσεις :

4.1 LoanWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο του δανείου. Ο constructor της είναι υπεύθυνος για την δημιουργία του παραθύρου που δίνει τις επιλογές δανείου στον χρήστη.

4.1.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) String loan
Περιγραφή : Η μεταβλητή που κρατάει την επιλογή του χρήστη.

4.1.2 Methods

a) public String getLoanOption()
Περιγραφή : Η μέθοδος αυτή επιστρέφει 0 αν ο παίκτης δεν επιλέξει να πάρει κάποιο δάνειο ή αν επιλέξει να πάρει δάνειο το ποσό του δανείου που θέλει ο παίκτης να πάρει.

4.2 LotteryWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο της θέσης Λοταρίας. Ο constructor της είναι υπεύθυνος για την δημιουργία του παραθύρου που δίνει την δυνατότητα στους παίκτες να ποντάρουν.

4.2.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) String bet
Περιγραφή : Η μεταβλητή που κρατάει το ποντάρισμα του παίκτη.

4.2.2 Methods

a) public String getBet()
Περιγραφή : Η μέθοδος αυτή επιστρέφει το ποντάρισμα του παίκτη. Αν ο παίκτης δεν έχει επιλέξει ο ίδιος ποντάρισμα παράγει έναν τυχαίο αριθμό και τον επιστρέφει ως ποντάρισμα του παίκτη.

4.3 MonthsToPlayWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο που θα καθορίσει πόσους μήνες επρόκειτο να διαρκέσει το παιχνίδι. Ο constructor της είναι υπεύθυνος για την δημιουργία του παραθύρου που θα δίνει την επιλογή στους παίκτες να διαλέξουν πόσους μήνες θα διαρκέσει το παιχνίδι.

4.3.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) String months
Περιγραφή : Η μεταβλητή που κρατάει τους μήνες που θα διαρκέσει το παιχνίδι.

4.3.2 Methods

a) public String getMonths()
Περιγραφή : Η μέθοδος αυτή επιστρέφει τους μήνες που επέλεξαν οι παίκτες να διαρκέσει το παιχνίδι. Αν οι παίκτες δεν επιλέξουν, τότε η μέθοδος επιστρέφει την τιμή 1 που θα είναι και οι μήνες που θα διαρκέσει το παιχνίδι.

4.4 PayDayWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο της PayDay. Ο constructor της είναι υπεύθυνος για την δημιουργία αυτού του παραθύρου.

4.4.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) int pChoice
Περιγραφή : Η μεταβλητή που κρατάει την επιλογή του παίκτη στην PayDay.

4.4.2 Methods

a) public int getPChoice()
Περιγραφή : Η μέθοδος αυτή επιστρέφει την επιλογή του παίκτη στην

PayDay.

b) public String partOfLoan()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την περίπτωση όπου ο παίκτης θέλει να πληρώσει ένα μέρος του δανείου του. Συγκεκριμένα, δημιουργεί ένα νέο παράθυρο, το οποίο δίνει επιλογές στον χρήστη και επιστρέφει την τιμή που ο χρήστης διάλεξε να πληρώσει.

4.5 SundayWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο της Κυριακής (για στοίχημα). Συγκεκριμένα, πέρα από τον αρχικό constructor περιέχει και έναν ακόμα constructor που δημιουργεί ένα παράθυρο με το αποτέλεσμα του bet στο ματς του El Clasico.

4.5.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) int bet
Περιγραφή : Η μεταβλητή που κρατάει την επιλογή του παίκτη για το ποντάρισμα.

4.5.2 Methods

a) public int getBet()
Περιγραφή : Η μέθοδος αυτή επιστρέφει την επιλογή του παίκτη για το ποντάρισμα.

4.6 ThursdayWindow

Πρόκειται για την κλάση που υλοποιεί το παράθυρο της Πέμπτης (για την αγορά κρυπτονομισμάτων). Συγκεκριμένα, πέρα από τον αρχικό constructor περιέχει και έναν ακόμα constructor που δημιουργεί ένα παράθυρο με το αποτέλεσμα του bet σε ένα κρυπτονόμισμα.

4.6.1 Attributes

a) ImageIcon image
Περιγραφή : Η φωτογραφία του παραθύρου.

b) int bet
Περιγραφή : Η μεταβλητή που κρατάει την επιλογή του παίκτη για το ποντάρισμα

4.6.2 Methods

a) int getBet()
Περιγραφή : Η μέθοδος αυτή επιστρέφει την επιλογή του παίκτη για το ποντάρισμα.

4.7 Window

Πρόκειται για μία βοηθητική κλάση η οποία θα έχει την γενική μορφή ενός παραθύρου και θα μας βοηθήσει να αντιστοιχίσουμε την δράση των κουμπιών με δεδομένα, ώστε να τα χρησιμοποιήσουμε για τις επιλογές των παικτών σχετικά με το μενού του παιχνιδιού. Ο constructor της είναι υπεύθυνος για την δημιουργία αυτού του παραθύρου.

4.7.1 Attributes

a) ImageIcon icon
Περιγραφή : Η φωτογραφία του παραθύρου.

b) int option
Περιγραφή : Η επιλογή του παίκτη στο μενού

c) int optionHelper
Περιγραφή : Βοηθητική μεταβλητή για την διαχείριση των ενεργειών σύμφωνα με την επιλογή του παίκτη.

4.7.2 Methods

a) public int getOption()
Περιγραφή : Η μέθοδος αυτή επιστρέφει την επιλογή του παίκτη στο μενού.

4.8 JLayeredPaneExtension

Πρόκειται για μία βοηθητική κλάση, η οποία μας επιτρέπει να τοποθετούμε τις εικόνες στο βασικό Panel του παιχνιδιού μας πιο εύκολα σε συνδυασμό με ένα inner class της View class που θα αναλύσουμε παρακάτω.

4.8.1 Attributes

a) private Image image
Περιγραφή : Η φωτογραφία που θέλουμε να τοποθετήσουμε.

4.8.2 Methods

a) public void paintComponent(Graphics g)
Περιγραφή : Η μέθοδος αυτή κληρονομείται από το super class και χρησιμοποιείται για την τοποθέτηση της εικόνας μας στο Panel του παιχνιδιού.

4.9 View

Πρόκειται για την βασική κλάση που υλοποιεί το UI του παιχνιδιού.

4.9.1 Attributes

a) Dimension screenSize
Περιγραφή : Χρησιμοποιείται για να πάρει το μέγεθος της κάθε οθόνης που

τρέχει το παιχνίδι.

b) private final int width
Περιγραφή : Χρησιμοποιείται για να σετάρει ένα fixed width που επεξεργαζόμαστε σε κάθε τι πρόκειται να εμφανιστεί, σύμφωνα με την ανάλυση της κάθε οθόνης που τρέχει το παιχνίδι.

c) private final int height
Περιγραφή : Χρησιμοποιείται για να σετάρει ένα fixed height που επεξεργαζόμαστε σε κάθε τι πρόκειται να εμφανιστεί, σύμφωνα με την ανάλυση της κάθε οθόνης που τρέχει το παιχνίδι.

d) private JLayeredPaneExtension panel
Περιγραφή : Το βασικό Panel του παιχνιδιού.

e) private URL imageURL
Περιγραφή : Χρησιμοποιείται για να κρατάει τη διεύθυνση κάθε εικόνας.

f) private Image image
Περιγραφή : Βοηθητική μεταβλητή για τις εικόνες των αντικειμένων του παιχνιδιού.

g) ImageIcon ins
Περιγραφή : Η μεταβλητή αυτή χρησιμοποιείται για να κρατάει εικόνες που θέλουμε να έχουμε στο Panel.

h) private JDesktopPane InfoBox
Περιγραφή : Χρησιμοποιείται για την δημιουργία του InfoBox στο Panel.

i) private JDesktopPane table
Περιγραφή : Χρησιμοποιείται για την δημιουργία του ταμπλό στο Panel.

j) private JDesktopPane p1
Περιγραφή : Χρησιμοποιείται για την δημιουργία της καρτέλας του 1ου παίκτη.

k) private JDesktopPane p2
Περιγραφή : Χρησιμοποιείται για την δημιουργία της καρτέλας του 2ου παίκτη.

l) private JDesktopPane jackpotPanel
Περιγραφή : Χρησιμοποιείται για την δημιουργία της θέσης του Jackpot στο ταμπλό.

m) private JLabel payDayImage
Περιγραφή : Χρησιμοποιείται για την τοποθέτηση του λογότυπου PayDay στο

ταμπλό.

n) private JLabel p1Name
Περιγραφή : Χρησιμοποιείται για την εμφάνιση του ονόματος του 1ου παίκτη.

o) private JLabel p2Name
Περιγραφή : Χρησιμοποιείται για την εμφάνιση του ονόματος του 2ου παίκτη.

p) private JLabel jackpotLabel
Περιγραφή : Χρησιμοποιείται για την τοποθέτηση της εικόνας του Jackpot στο ταμπλό.

q) private JButton dealCards
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού των καρτών «Συμφωνίας».

r) private JButton mailCards
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού των καρτών «Μηνύματος».

s) private JButton DealCards1
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «My Deal Cards» του 1ου παίκτη.

t) private JButton DealCards2
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «My Deal Cards» του 2ου παίκτη.

u) private JButton getLoanP1
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «Get Loan» του 1ου παίκτη.

v) private JButton getLoanP2
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «Get Loan» του 2ου παίκτη.

w) private JButton endTurnP1
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «End Turn» του 1ου παίκτη.

x) private JButton endTurnP2
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού «End Turn»

του 2ου παίκτη.

y) private JButton diceP1
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού που αποτελεί το ζάρι του 1ου παίκτη.

z) private JButton diceP2
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του κουμπιού που αποτελεί το ζάρι του 2ου παίκτη.

aa) private JTextField balance1Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζεται το υπόλοιπο του παίκτη 1.

bb) private JTextField balance2Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζεται το υπόλοιπο του παίκτη 2.

cc) private JTextField loan1Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζονται τα δάνεια του παίκτη 1.

dd) private JTextField loan2Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζονται τα δάνεια του παίκτη 2.

ee) private JTextField bills1Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζονται οι λογαριασμοί του παίκτη 1.

ff) private JTextField bills2Text
Περιγραφή : Χρησιμοποιείται για να εμφανίζονται οι λογαριασμοί του παίκτη 2.

gg) private JTextField jackpotText
Περιγραφή : Χρησιμοποιείται για να εμφανίζεται το ποσό του Jackpot.

hh) private JTextField info
Περιγραφή : Χρησιμοποιείται για να εμφανίζει το κείμενο “Info Box”.

ii) private JTextField monthsLeft
Περιγραφή : Χρησιμοποιείται για να εμφανίζει το πόσοι μήνες απομένουν στο παιχνίδι.

jj) private JTextField command
Περιγραφή : Χρησιμοποιείται για να εμφανίσει το γεγονός που έχει συμβεί τελευταίο στο παιχνίδι.

kk) private JTextField temp
Περιγραφή : Βοηθητική μεταβλητή για την αρχικοποίηση των θέσεων στο

ταμπλό.

ll) private JMenu menu
Περιγραφή : Χρησιμοποιείται για την υλοποίηση του Μενού του παιχνιδιού.

mm) private JDesktopPane[] position
Περιγραφή : Πρόκειται για τον πίνακα που περιέχει τις θέσεις που βρίσκονται επάνω στο ταμπλό.

nn) JLayeredPane[] pawn_position
Περιγραφή : Πρόκειται για τον πίνακα που περιέχει τις θέσεις που βρίσκονται οι παίκτες στο παιχνίδι.

oo) int choiseP1
Περιγραφή : Χρησιμοποιείται για την επιλογή του πονταρίσματος του παίκτη 1 στη θέση «Λοταρίας».

pp) int choiseP2
Περιγραφή : Χρησιμοποιείται για την επιλογή του πονταρίσματος του παίκτη 2 στη θέση «Λοταρίας».

qq) private int counter
Περιγραφή : Βοηθητικός μετρητής.

rr) private boolean radioPos
Περιγραφή : Χρησιμοποιείται για να δηλώσει αν μία μέρα στο ταμπλό είναι «Διαγωνισμός στο Ραδιόφωνο» (true) ή όχι (false).

ss) private boolean lotteryPos
Περιγραφή : Χρησιμοποιείται για να δηλώσει αν μία μέρα στο ταμπλό είναι «Θέση Λοταρίας» (true) ή όχι (false).

tt) private ClassLoader cldr
Περιγραφή : Χρησιμοποιείται για να φορτώνει δυναμικά κλάσεις στην εικονική μηχανή της Java (JVM).

uu) Controller controller
Περιγραφή : Χρησιμοποιείται έτσι ώστε μέσω του controller να ανανεώνεται κατάλληλα η εξέλιξη του παιχνιδιού.

4.9.2 Methods

a) public void initComponents()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την δημιουργία των βασικών στοιχείων του παιχνιδιού (από γραφικής μεριάς). Συγκεκριμένα, τις στοίβες όλων των καρτών, του λογότυπου του παιχνιδιού στο πάνω μέρος της οθόνης, του αρχικού Panel πάνω στο οποίο βρίσκονται όλα τα γραφικά

στοιχεία του παιχνιδιού, το background του Panel, τις καρτέλες των 2 παικτών και την μπάρα του μενού.

b) private void start()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την έναρξη ενός παιχνιδιού. Συγκεκριμένα μέσω του controller αρχικοποιεί τα στοιχεία του παιχνιδιού και στη συνέχεια δημιουργεί το ταμπλό του παιχνιδιού, τα πιόνια των παικτών, τα ζάρια και το InfoBox.

c) public void newGame()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την αρχικοποίηση ενός νέου παινχιδιού.

d) public void setInfobox()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να κάνει το InfoBox χρήσιμο, καθώς καθορίζει το μέγεθος και τη θέση του επάνω στο ταμπλό και προσθέτει τα σημεία στα οποία παρέχει χρήσιμες πληροφορίες κατά τη διάρκεια του παιχνιδιού.

e) public void boardPositions()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την αρχικοποίηση του ταμπλό και των θέσεων του.

f) public void menuInitialize()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την δημιουργία του μενού του παιχνιδιού.

g) public void playerPanel()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την δημιουργία των καρτελών και των 2 παικτών.

h) public void mailCardGfxAction(Player p, MailCard c)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την υλοποίηση των γραφικών ενεργειών που πρέπει να κάνει κάθε κάρτα «Μηνύματος». Επιπλέον, δέχεται ως όρισμα τον παίκτη πάνω στον οποίο γίνεται η ενέργεια της κάρτας, καθώς και την κάρτα μηνύματος που θα χρειαστεί για την εύρεση της ενέργειας που πρέπει να πραγματοποιηθεί σύμφωνα με τον τύπο της.

i) public void Sunday(Player p)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση του παραθύρου Κυριακής (Ποδοσφαιρικού αγώνα). Ανάλογα την επιλογή του παίκτη στη συνέχεια εμφανίζει το παράθυρο με το κατάλληλο μήνυμα. Επιπλέον, δέχεται ως όρισμα τον παίκτη, ο οποίος βρίσκεται εκείνη τη στιγμή σε θέση που η μέρα είναι Κυριακή.

j) public void Thursday(Player p)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση του παραθύρου Πέμπτης (Crypto Thursday). Στη συνέχεια ανάλογα με την επιλογή του παίκτη εμφανίζει το παράθυρο με το κατάλληλο μήνυμα. Επιπλέον, δέχεται ως όρισμα τον παίκτη, ο οποίος βρίσκεται εκείνη τη στιγμή σε θέση που η μέρα είναι Πέμπτη.

k) public void playerMove(ActionEvent e)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να μετακινούνται τα πιόνια των παικτών επάνω στο ταμπλό. Επιπλέον, δέχεται ως όρισμα ένα event που έγινε triggered από το ζάρι.

l) private void positionGraphics(Player p, int position)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την υλοποίηση των γραφικών ενεργειών που πρέπει να εκτελέσει κάθε θέση. Επιπλέον, δέχεται ως όρισμα τον παίκτη, ο οποίος βρίσκεται σε αυτή τη θέση και έναν ακέραιο που δηλώνει το δείκτη της θέσης κάθε φορά.

m) public void getLoan(Player p)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση του παραθύρου «Δανείου» και για τις γραφικές ενέργειες που απαιτούνται όταν ένας παίκτης παίρνει κάποιο δάνειο.

n) public void playerDealCards(Player p)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για τις ενέργειες που πρέπει να γίνουν όταν ένας παίκτης πουλάει μία κάρτα «Συμφωνίας». Επιπλέον, δέχεται ως όρισμα τον παίκτη που πρόκειται να πουλήσει την κάρτα.

o) private void paintPlayer()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να εμφανίζονται τα βασικά στοιχεία του κάθε παίκτη στην καρτέλα του.

p) public void paintPawn(String p, int pos)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να εμφανίζονται τα πιόνια κάθε παίκτη στο ταμπλό. Επιπλέον, δέχεται ως όρισμα το όνομα κάθε παίκτη, καθώς και τον δείκτη της θέσης στην οποία βρίσκεται ο παίκτης πάνω στο ταμπλό.

q) public void paintJackpot()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να εμφανίζονται τα λεφτά

που υπάρχουν στο Jackpot του παιχνιδιού τη δεδομένη στιγμή.

r) private void paintInfoBox(String text)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να εμφανίζονται οι πληροφορίες στο InfoBox. Επιπλέον, δέχεται ως όρισμα ένα String το οποίο πρόκειται για την εντολή που εκτελείται τη δεδομένη στιγμή στο παιχνίδι.

s) private void paintDice(String p, int diceNumber)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την γραφική απεικόνιση και την γραφική ανανέωση του ζαριού.

t) public int showDealCard(DealCard C, String option)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την υλοποίηση του παραθύρου «Κάρτας Συμφωνίας». Επιπλέον, δέχεται ως όρισμα μία κάρτα συμφωνίας, την οποία πρόκειται να εμφανίσει και την επιλογή του παίκτη για το εάν θα αγοράσει ή όχι την κάρτα αυτή.

u) public void showMailCard(Card c)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για την υλοποίηση του παραθύρου «Κάρτα Μηνύματος». Επιπλέον, δέχεται ως όρισμα μία κάρτα μηνύματος έτσι ώστε να εξακριβώσει στη συνέχεια το είδος της κάρτας και να εμφανίσει το κατάλληλο παράθυρο με τα κατάλληλα στοιχεία.

v) static ImageIcon getImageScaled(String image, int width, int height)
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για να μπορέσουμε να πάρουμε όσο πιο ομαλά scaled τις εικόνες στο παιχνίδι.

w) public void startingPlayer()
Περιγραφή : Η μέθοδος αυτή χρησιμοποιείται για τον καθορισμό του παίκτη που θα παίξει πρώτος όταν ξεκινάει το παιχνίδι.

4.9.3 Inner Classes

* Τα Listeners στην περιεγράφηκαν ως τμήμα του Controller. Ωστόσο, επειδή συμβάλουν στην ανανέωση του γραφικού περιβάλλοντος του παιχνιδιού και για την τήρηση του μοντέλου ανάπτυξης MVC, θεώρησα πως η θέση τους θα πρέπει να βρίσκεται στο View.

a) *private class cardListener implements ActionListener
Περιγραφή : Η κλάση cardListener κάνει implement το interface ActionListener και υλοποιεί την μέθοδο που κληρονομεί και υλοποιεί (public void actionPerformed(ActionEvent e)) για την αντιμετώπιση των συμβάντων από το γραφικό επίπεδο των καρτών.

b) *private class buttonListener1 implements ActionListener
Περιγραφή : Η κλάση buttonListener1 κάνει implement το interface ActionListener και υλοποιεί την μέθοδο που κληρονομεί και υλοποιεί (public void actionPerformed(ActionEvent e)) για την αντιμετώπιση των συμβάντων από το γραφικό επίπεδο του 1ου παίκτη του παιχνιδιού.

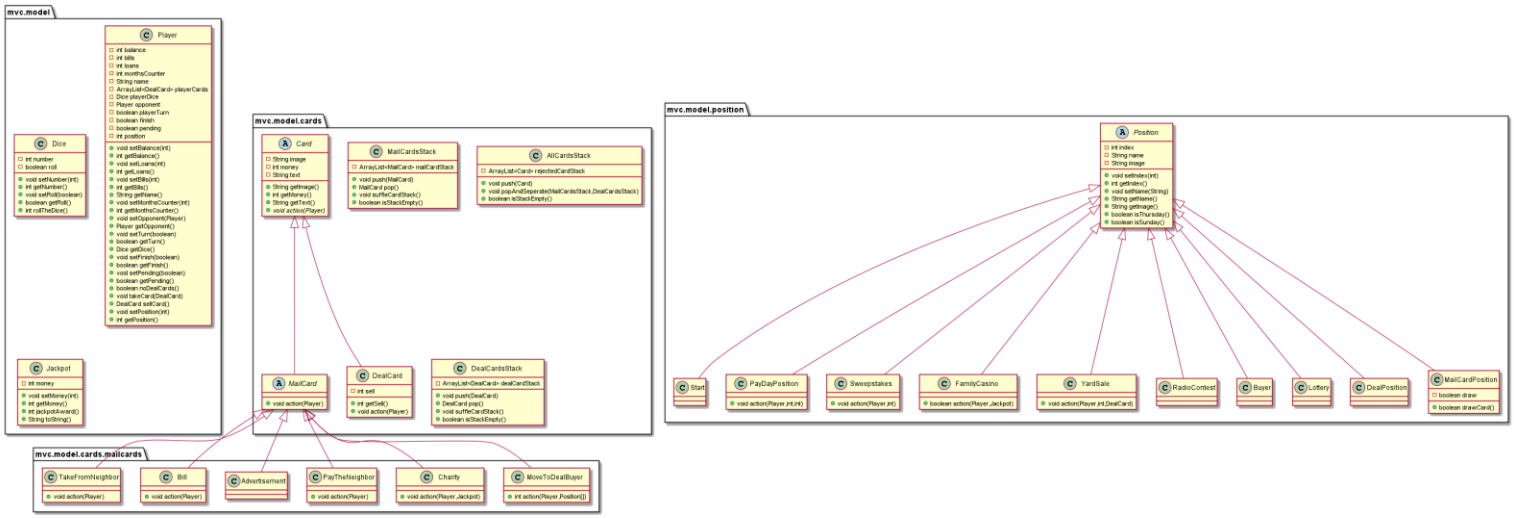
c) *private class buttonListener2 implements ActionListener
Περιγραφή : Η κλάση buttonListener2 κάνει implement το interface ActionListener και υλοποιεί την μέθοδο που κληρονομεί και υλοποιεί (public void actionPerformed(ActionEvent e)) για την αντιμετώπιση των συμβάντων από το γραφικό επίπεδο του 2ου παίκτη του παιχνιδιού.

d) *private class menuListener implements ActionListener
Περιγραφή : Η κλάση menuListener κάνει implement το interface ActionListener και υλοποιεί την μέθοδο που κληρονομεί και υλοποιεί (public void actionPerformed(ActionEvent e)) για την αντιμετώπιση των συμβάντων από το γραφικό επίπεδο του μενού του παιχνιδιού.

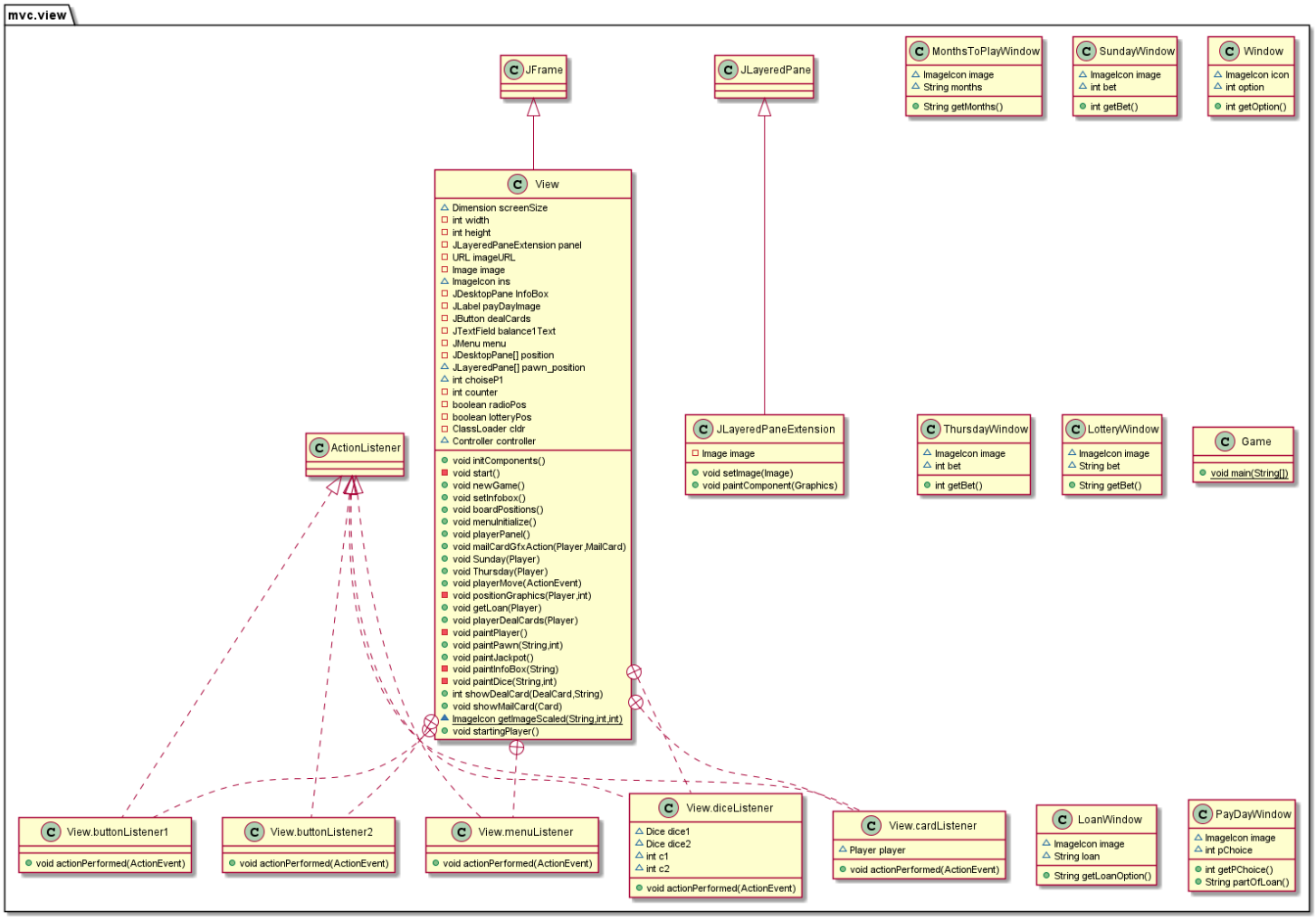
e) *private class menuListener implements ActionListener
Περιγραφή : Η κλάση menuListener κάνει implement το interface ActionListener και υλοποιεί την μέθοδο που κληρονομεί και υλοποιεί (public void actionPerformed(ActionEvent e)) για την αντιμετώπιση των συμβάντων από το γραφικό επίπεδο του ζαριού του παιχνιδιού.

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

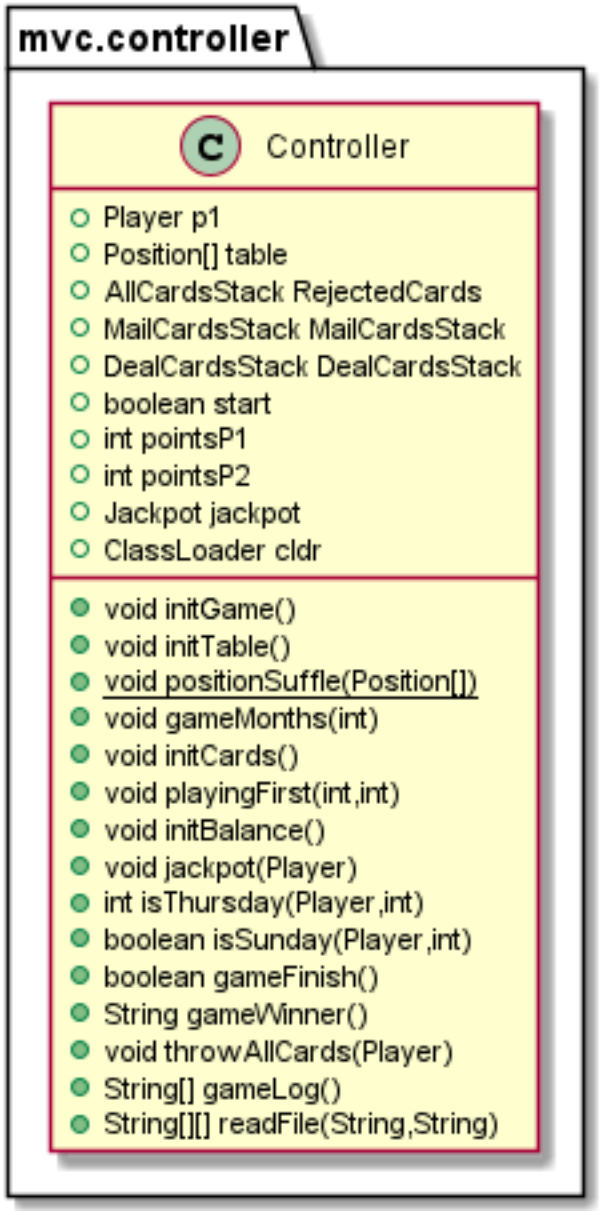
Model UML Diagram



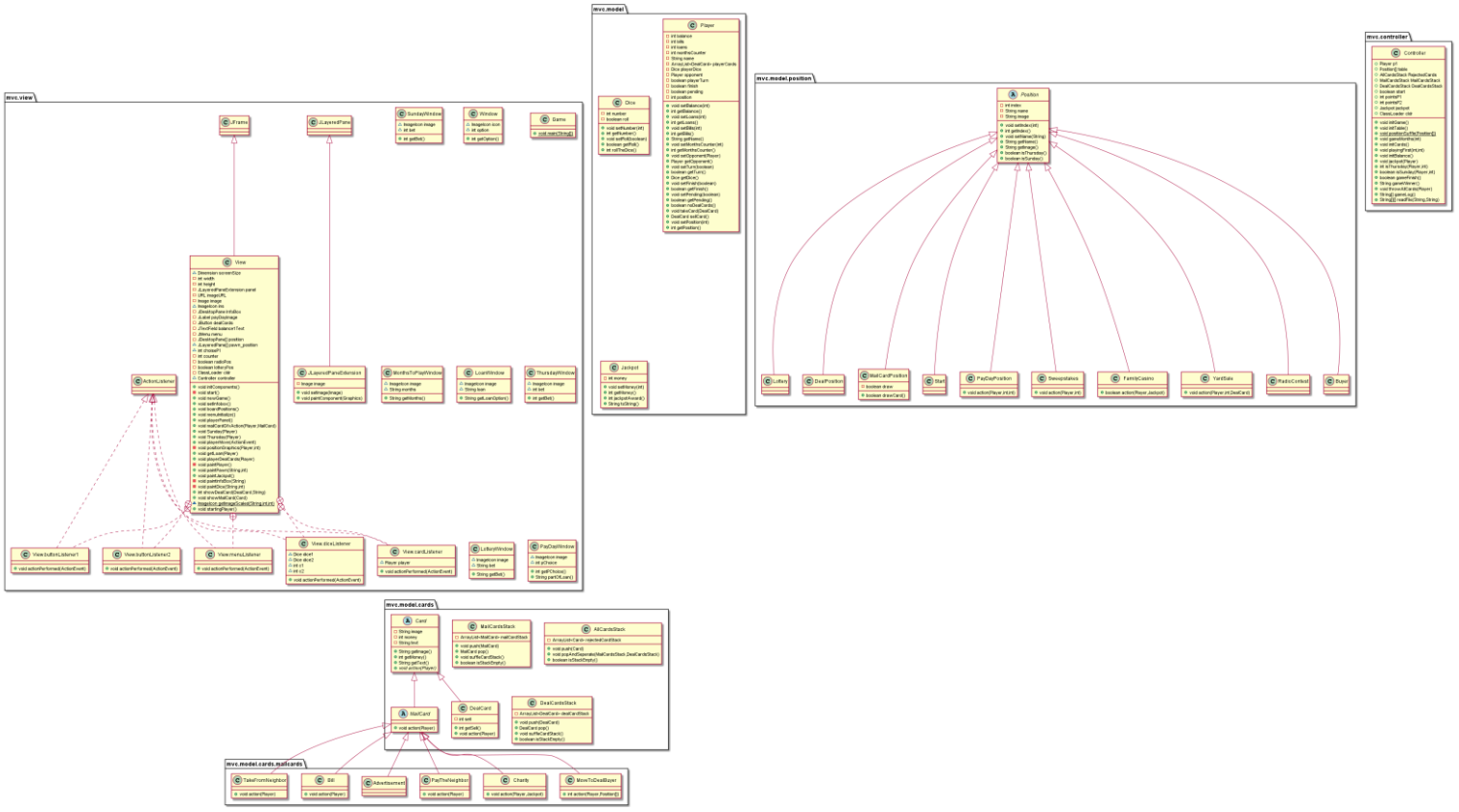
View UML Diagram



Contoller UML Diagram



MVC UML Diagram



Λόγω του ότι τα διαγράμματα είναι αρκετά μεγάλα και μπορεί να μην είναι τόσο ευδιάκριτα στην αναφορά, επισυνάπτονται μαζί και τα σχετικά αρχεία.

Αυτό που διακρίνουμε από τα διαγράμματα UML είναι ότι, παρατηρούμε με τα βελάκια που ξεκινούν από κάποια κλάση να δείχνουν την κλάση την οποία κάνουν extend, δηλαδή που δηλώνουν την έννοια της κληρονομικότητας μεταξύ των κλάσεων. Επιπλέον, παρατηρούμε ένα σταυρό μέσα σε έναν κύκλο που δηλώνει ότι οι κλάσεις αυτές που δείχνουν με αυτό το σύμβολο προς μία άλλη κλάση είναι εσωτερικές κλάσεις (π.χ. cardListener είναι inner class της class Controller).

6. Λειτουργικότητα (Β Φάση)

Θεωρώ πως υλοποιήθηκαν όλα τα ερωτήματα επιτυχώς. Δυστυχώς υπήρχε δυσκολία στην δημιουργία tests και συνεπώς δεν υλοποιήθηκαν.

7. Συμπεράσματα

Το θέμα της εργασίας ήταν πολύ ενδιαφέρον και χρήσιμο για να συνηθίσουμε να γράφουμε σε Java, αλλά και για το πως είναι να δουλεύει κανείς σε ένα μεγάλο (για αρχή) project. Σίγουρα μάθαμε αρκετά πράγματα για την γλώσσα και για το πως πρέπει να οργανώνουμε τον κώδικα μας πάνω σε ένα συγκεκριμένο μοντέλο ανάπτυξης. Κατά την άποψη μου η χρήση Java Swing δυσχεραίνει την ανάπτυξη ενός προγράμματος στο μοντέλο MVC καθώς επίσης καθυστερεί την διαδικασία ανάπτυξης της εφαρμογής γιατί δεν βλέπουμε συγχρόνως το γραφικό κομμάτι του παιχνιδιού όταν το επεξεργαζόμαστε.