

Πανεπιστήμιο Κρήτης - Τμήμα Επιστήμης Υπολογιστών

Έτος 2021-22 , Χειμερινή περίοδος

HY252-Αντικειμενοστραφής Προγραμματισμός

Διδάσκων : Ι. Τζίτζικας

Ηλίας Καψής - csd4652

ΠΡΟΤΖΕΚΤ

Εισαγωγή :

Σκοπός αυτής της άσκησης , είναι ο σχεδιασμός και η υλοποίηση του διάσημου παιχνιδιού Payday στα Ελληνικά . Σε αυτή τη παρουσίαση θα αναλύσω την σκέψη πίσω από τον σχεδιασμό του παιχνιδιού και θα επισημάνω τις διάφορες κλάσεις και μεθόδους που χρησιμοποιώ.

Περιεχόμενα

- 1. Εισαγωγή 2**
- 2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model –UML MODEL 2**
- 3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller–UML MODEL 10**
- 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View –UML MODEL 17**
- 5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 21**
- 6. Λειτουργικότητα + Σημείωση 23**

Εισαγωγή (vol 2):

Αρχικά , έπρεπε να χωρίσω το πρόγραμμα μου στο μοντέλο MVC. Δηλαδή, έχω χωρίσει τα γραφικά του παιχνιδιού από τον βασικό πυρήνα του. Οπότε ο χρήστης βλέπει το view, με το controller αλλάζει τις ιδότητες του model , τα οποία στην συνέχεια αλλάζουν και το view. Έτσι , τα packages model και view για να αλληλεπιδράσουν χρειάζονται την μεσολάβηση του Controller η οποία εξαιτίας αυτής της ικανότητας που έχει , μπορούμε να την καλεσούμε σαν τον εγκεφαλο του παιχνιδιού.

Package Model:

Το package model θα περιέχει τις κλάσεις :

- Class Position
- Abstract class player
- Class controller extends player
- Abstract Class Tile
- MessageTile extends Tile
- AgreementTile extends Tile
- Class Board
- Class Dice

Abstract Class Player

Αρχικά φτιάχνουμε μια abstract class Player η οποία θα έχει

Τα attributes:

Private Int money;

Private Int loanedcash;

Private int bills;

Private String name;

Private int month;

Dice roll = new Dice();

Position cords = new Position();

Stack<AgreementCards> stack = new Stack<AgreementCards>();

Οι μέθοδοι:

Player(String x); //constuctor that calls set

Set(); //sets money to 3500 and loanedcash to 0, month=0, and
cords.board[]=0

Class Controller extends Player

Θα έχει τα attributes της inhereted κλάσης Player και τα methods

CharacterString x)

Void move(int x) αλλάζει τη θέση του παίκτη

Int RollDice() επιστεφει ακαιρειο από 1 εως 6

Void changeMoney(int x) αλλάζει χρηματα κατα x

Void changeLOAN(int x) παινει x\$ δανειο

Void changeBills(int x) αλλάζει το συνολικο ποσο απληρωτων λογαριασμων

Int GetMoney(){return super.money}

Int getloaned(){return super.loanedcash}

Int get bills(){return super.bills}

Void giveMoney(Character b, int x) δινει στο b x ποσο χρηματων

Void paybills() αποπληρωνει το χρεος του

Boolean hasFinished(Board v) ελεγχει αν ο παικτης έχει τελειώσει το
παιχνιδι

Void buy(int x) αγοράζει και μειώνει τα χρήματα του παίκτη κατά x

Void adjust(int x) Αυτομάτα αν τα χρήματα του παίκτη δεν αρκούν για να αποπληρωσει το ποσό x δανειζονται αρκετά χρήματα στον παίκτη για να ολοκληρωθεί η αγορά.

Class Position

Αυτή η κλάση θα δείχνει σε ποια θέση του board βρίσκεται ο player και θα η οποία μπορεί να φτάσει μέχρι 35

Τα attributes είναι :

Int x ακέραιος που θυμάται την θέση κάποιου object

Οι μέθοδοι είναι :

Void method setPosition(int x, Character c){} αυτή η μέθοδος έχει σχεδιαστεί έτσι ώστε όταν το cords πάει >30 να πηγαίνει στην θέση 30 η οποία είναι το PayDayTile του board μας.Επίσης αν είμαστε ήδη στην θέση 30, προσθετούμε το αριθμό ζαριού με το -1 επειδή ξέρουμε ότι οι πίνακες στην java αρχίζουν από 0 αρα $-1+1(\text{min number of dice}) = 0$.

Class Dice

Αυτή η κλάση θα έχει όριο από 1 μέχρι 6.

Attributes:

Int numb=0;

METHODS:

Int roll(){ επιστρεφει αριθμο 1 εως 6

Int getDice() επιστρεφει τον αριθμο που είναι αποθηκευμενος στο x

CLASS BOARD

Αυτή η κλάση δημιουργει το board του παιχνιδιού οπου αρχικοποιουνται οι καρτες και τα πλακιδια του παιχνιδιου.

Τα atributes :

public Tile[] MyBoard = new Tile[31]; Πινακας των πλακιδιων που είναι instance Tile

public int time; Οι μηνες που θα διαρκεσει το παιχνιδι

AgreementCards[] AT = new AgreementCards[20]; Πινακας που αποθηκευονται ολες οι καρτες συμφωνιας του παιχνιδιου με τα instanes τους

MailCards[] MT = new MailCards[48];

Πινακας που αποθηκευονται ολες οι καρτες μνηματος του παιχνιδιου

String[][] mailCards = new String[48][4]; Πινακας που αποθηκευει τα στοιχεια των καρτων μνηματος

Int jackpotamount Είναι το συνολο των χρηματων που εχουν μαζευτει στο jackpot Tile

Int currentcard Αυτος ο αριθμος δειχνει σε ποια καρτα συμφωνιας βρισκομαστε. Η χρηση της είναι πολύ σημαντικη καθως αντι να αποβαλω κάθε καρτα που αγοραζει ο χρηστης μολις την αγορασει καποιος την προσπαιρνω.Οταν τελειωσουν ολες αρχιζω από το 0.

int currentmail Αυτος ο αριθμος δειχνει σε ποια καρτα μνηματος βρισκομαστε. Η χρηση της είναι πολύ σημαντικη καθως αντι να αποβαλω κάθε καρτα που αγοραζει ο χρηστης μολις την αγορασει καποιος την προσπαιρνω.Οταν τελειωσουν ολες αρχιζω από το 0.

`String[][] dealCards = new String[20][8];` Πίνακας που αποθηκεύει τα στοιχεία των καρτών συμφωνίας

Οι μέθοδοι :

`Board(int x)` Δημιουργεί το board και αρχικοποιεί τους μήνες που θα διαρκέσει το παιχνίδι

`void initTiles()` Αρχικοποιεί τις κάρτες του παιχνιδιού

`void initBoard()` Αρχικοποιεί τα πλακίδια του παιχνιδιού

`CheckEmpty(int x, Tile c)` Ελέγχει κενές θέσεις και βάζει σε τυχαία σειρά τα πλακίδια του παιχνιδιού

`void readFile(String path, String type)` Διαβάζει από το csv αρχείο τα attributes κάθε κάρτας

`Void randomiseCards()` Αυτή η μέθοδος ανακατεύει τις κάρτες του παιχνιδιού

Interface CARDS

Αυτό το interface αντιπροσωπεύει τις δυνατότητες κάποιας κάρτας και δίνει κάποιες ειδικές μεθόδους:

`void get(Character r);` είναι ακόμα unimplimented

`void move(Character r);` κουνάει το παίκτη σε κάποιο πλακίδιο του board (Mail card ability)

`void pay(Character r, Character b);` ο παίκτης r πληρώνει ένα ποσό τον παίκτη b (Mail card ability)

`void getpayed(Character r, Character b);` ο παίκτης b πληρώνει ένα ποσό τον παίκτη r (Mail card ability)

`void payBank(Character r);` ο παίκτης πληρώνει ένα ποσό στην τραπεζα

(Mail card ability)

void billCard(Character r); ο λογαριασμός του παίκτη χρεώνεται κάποιο ποσό
(Mail card ability)

void buy(Character r); ο παίκτης αγοράζει κάποια κάρτα (Deal card ability)

void sell(Character r); ο παίκτης πουλάει κάποια κάρτα (Deall card ability)

Void getcard(int temp Character c); Δίνει στον χρήστη την διαθέσιμη κάρτα συμφωνίας και την προσθέτει στο inventory του

CLASS MailCards implementsCards

Αυτή η κλάση εξελίζει την κλάση cards και αντιπροσωπεύει κάρτες μνημάτων του παιχνιδιού .

Τα attributes:

int value; το ποσό που εκδηλώνει η κάρτα

String image; η φωτογραφία της κάρτας

String message; το μήνυμα της κάρτας

String type; ο τύπος της κάρτας

Τα methods περιγράφονται αναλυτικά στην Cards

CLASS AgreementCards implements Cards

Αυτή η κλάση εξελίζει την κλάση cards και αντιπροσωπεύει κάρτες συμφωνίας του παιχνιδιού .

Τα attributes:

int value; το ποσό που αξίζει η κάρτα για αγορά

String image; η φωτογραφία της κάρτας

String message; το μήνυμα της κάρτας

Int sell value το ποσο που αξίζει η πωληση της καρτας

Τα methods περιγραφονται αναλυτικα στην Cards

Abstact CLASS Tile

Αυτή η κλάση αντιπροπευει ένα γενικευμενο πλακιδιο του παιχνιδιου. Αυτη χωριζεται σε πολλες υποκλασεις οι οποιες δεν εχουν καποιο function αλλα χρησιμοποιουνται για την αναγνωριση ποιας ενεργειας πρεπει να ακολουθησει ο παικτης αφου βρισκεται πανω σε ένα συγκεκριμενου ειδους πλακιδιου.

ΟΛΕΣ ΟΙ ΥΠΟΚΛΑΣΕΙΣ:

Class BuyerTile

Class CasinoTile

Class DealTILE

Class JackPotTile

Class Lottery Tile

ClassMail Tile

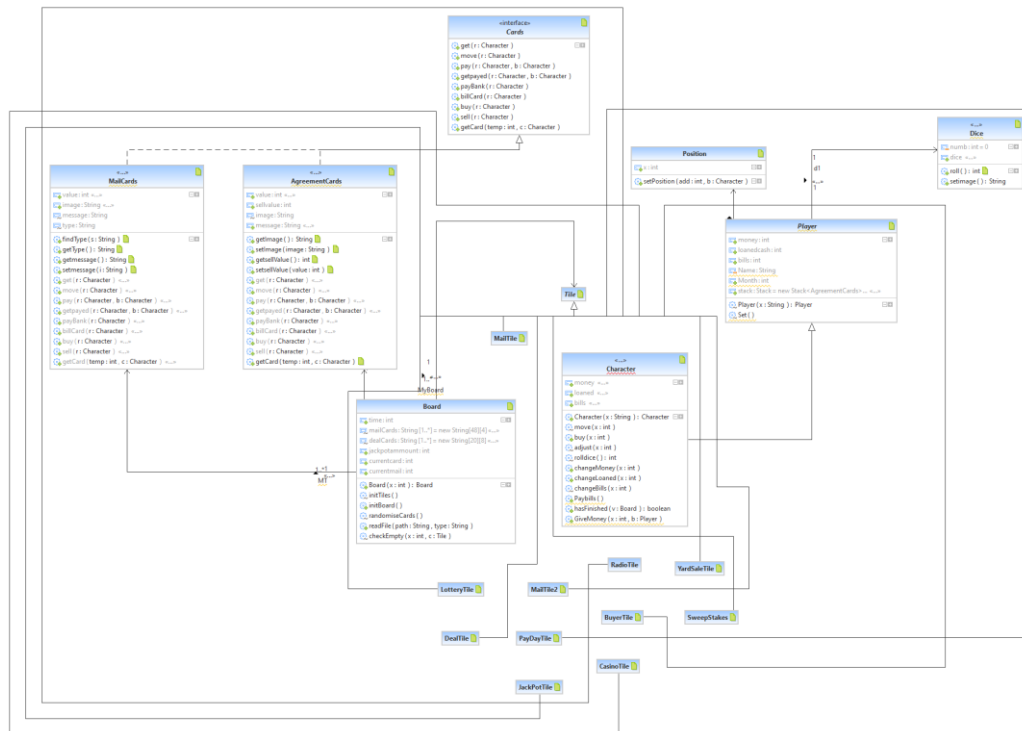
ClassMail Tile2

Class PayDay Tile

Class Radio Tile

Class SweepStakes Tile

Class YardSale Tile



Αυτό είναι το Uml Class Diagramm του Model. Η κλάση character κάνει extend την κλάση Player αρα παίρνει όλα τα attributes και τις μεθοδους της. Ειδικότερα, η player έχει επίσης σαν attribute το dice ώστε να μπορεί να ριχτεί ζαρί και Position για να βρισκουμε σε ποια θέση του board είναι. Οι κλάσεις που συμβολίζουν τις κάρτες συμφωνίας και μνημάτων του παιχνιδιού μας (οι MailCards και οι Agreement cards) κάνουν extend το interface Cards το οποίο έχει πολλές μεθοδους. Τέλος, η κλάση Board που αναπαριστά το ταμπλό του παιχνιδιού. Η οποία συμπεριλαμβάνει μέσα τις κάρτες μνημάτων, συμφωνίας και κάθε είδος ειδικού πλακιδίου του παιχνιδιού.

Package Controller:

Το package controller έχει ως σκοπο να είναι ο εγκεφαλος του παιχνιδιου.

Δηλαδή είναι εκείνο το οποίο θα επιτρέπει το view και το model να επικοινωνούν(αλληλεπιδρούν) και να προχωράνε το παιχνίδι.Το controller καταφέρνει να αρχικοποιεί το παιχνίδι , αφήνει τους παίκτες να παίξουν την σειρά τους και στο τέλος τελειώνει το παιχνίδι και ανακοινώνει τον νικητή.Αρα θα χρειαστεί κάποιες μεθόδους και attribute που επιτελούν να το κάνει αυτό.

CLASS Controller

Κλάση η οποία ελέγχει την ροή του παιχνιδιου

Τα attributes :

Static int x; Αυτή η μεταβλητή είναι χρησιμη για τον υπολογισμό της σειράς του παιχνιδιου

Static view Myview; Είναι attribute της κλάσης γραφικών που εφτιαξα η οποία φροντίζει να απεικονισει σε παραθυρο το board μας .

Static int turn : Ελεγχει την σειρά του παιχνιδιου.

Static int fr1; : Χρησιμοποιειται σε περιπτώσεις οπου ένας από τους χρηστες βρισκεται σε radio Tile και πρεπει να ριξουν ξανα ζαρι

Static int fr2; : Το ιδιο με fr1 απλως για τον άλλο χρηστη

Static int decision; χρησιμοποιηται για όταν ο χρηστης βρισκεται σε μερα Κυριακη και Πευπη και πρεπει να κανει μια αποφαση .

Static int AmmountMail ; Είναι χρησιμο για όταν ο χρηστης πρεπει να παρει καρτες Μυνηματος αναλογα σε ποιο tile πεσει γινεται 1 ή 2, αναλογα αν είναι Maletile ή Mailetile2.

Static int yesfroll ; Πολύ σημαντικό γιατι αυτό ξεχωριζει την διαφορα μεταξυ κανονικης ριξης ζαριους και ψευτικη . Όταν είναι true ο χρηστης μπορεί να ριξει ζαρι χωρις να κουνησει από την θεση του .

Static boolean p1hasplayed; Επίσης πολύ σημαντικό όταν ο χρηστης πατησει end αυτό γινεται true και μπορεί να πατησει τα κουμπια του καθώς είναι σειρά του αλλου παικτη . Κανονικα το turn θα αρκουσε αλλα επειδη σε καποια tiles ο παικτης πρεπει να ξαναριξει ζαρι μια τετοια μεταβλητη είναι αναγκαια.

static boolean p2hasplayed ; το ιδιο με πριν απλα για τον άλλο παικτη

static Character player1; ο πρωτος παικτης

static Character player2; ο δευτερος παικτης

Static string in χρησιμοποιειται σπανια για να κανει update το Text μας

public static Board b1; Το board του παιχνιδιου μας που δημιουργησαμε στο Model. Αναγκαιο για την δημιουργεια της πλακετας του παιχνιδιου.

public static boolean mgo; Χρησιμοποιηται για να απογορευει τον χρηστη να σηκωνει καρτες μνηματος χωρις να βρισκεται πανω στο ειδικο tile τους.Γινεται true όταν παιφτει πανω σε mail tile 1 ή 2.

public static boolean dgo; χρησιμοποιηται για να απογορευει τον χρηστη να σηκωνει καρτες μνηματος χωρις να βρισκεται πανω στο ειδικο tile τους.Γινεται true όταν παιφτει πανω σε DealTile

static boolean Sunday; χρησιμοποιηται για όταν ο Χρηστης πεσει σε μερα Κυριακη και όταν γινεται αληθης επιτρεπει στον χρηστη να κανει την καταλληλη πραξει

static boolean Thursday; χρησιμοποιηται για όταν ο Χρηστης πεσει σε μερα Περμπτη και όταν γινεται αληθης επιτρεπει στον χρηστη να κανει την καταλληλη πραξει

static boolean p1; όταν γινεται αληθης επιτρεπει στον παικτη να πατησει το κουμπι inventory και να πουλησει καρτα

static boolean p2; ίδιο με πριν

Οι μεθοδοι :

private static void controller() Δημιουργει τον 1ο εγκεφαλο του παιχνιδιου και αρχικοποιει την σειρα του παιχνιδιου το board μας , τα γραφικά μας και τα ενημερωτικά σχολια.

static int selectTime() Αρχικοποιει την διαρκεια του παιχνιδιου

static int calculate(Character p) Υπολογιζει το σκορ του παικτη

static boolean hasGameFineshed(Character a, Character b,Board c)

Ελεγχει αν το παιχνιδι εχει τελειωσει

Static int calulateTurn() αποφασιζει την σειρα των παικτων

Void updateText() and updateText(String S)και οι 2 εχουν σκοπο το να ανανεωνουν τα ενημερωτικά μνηματα του παιχνιδιου.

Void play(character c, int x) Ο δευτερος εγκεφαλος του παιχνιδιου και παρα πολύ σημαντικός.Πηγαινει στην θεση που βρισκεται ο παικτης και βρισκει αναλογα σε ποιο Tile ΚΑΙ!! ποια μερα που είναι την καταλληλη πραξη και αν τελειωσε ο παικτης που αν τελειωσε κανει σιγουρο ότι δεν αλλαζει η σειρα.

Void roll1() το ζαρι του πρωτου χρηστη και μετακινει τον παικτη στη θεση που εριξε το ζαρι του. Επισης ελεγχει αν εριξε εξι αν τελειωσε το παινιδι και αν επαισε σε μερα Κυριακη η Περμπτη.

Void roll2() ίδιο με το προηγουμενο απλως είναι του αλλου χρηστη

`Static void fakeroll()` ; 3ος εγκεφαλός παιχνιδιού αναλαμβάνει τις πράξεις που πρέπει ο χρήστης να ξαναριξεί ζάρι όπως πχ `sweepstake` , `Sunday` , `Thursday` κτλπ, και τα στέλνει πίσω στο `play(character x,int x)` ή στην κατάλληλη συνάρτησή τους. Είναι αναγκαία επειδή αν δεν υπήρχε και ήταν μόνο το `roll` το πρόγραμμα θα έκανε τις πράξεις χωρίς να προλάβει ο χρήστης να ξαναριξεί το ζάρι. Αυτή η μέθοδος κάνει σίγουρο ότι πρώτα θα ρίξει ζάρι ο χρήστης και μετά θα πάει στη κατάλληλη συνάρτηση .

`Static void end1()`; Αυτή η μέθοδος τελειώνει την σειρά του παίκτη 1 αφού έχει ολοκληρώσει τις πράξεις του. Επίσης τσεκάρει αν έχουν τελειώσει και οι 2 παίκτες την σειρά τους όπου μετά το ξαναβαάζει στην αρχική σειρά τους .

`Static void end2()` Ιδίο με `end1()`

`Static void inventory1()` ; ανοίγει το `inventory` του παίκτη και βλέπει ποιες κάρτες έχει αγοράσει . Επίσης έχει την δυνατότητα να πουλήσει κάρτες.

`Static void inventory2()` ; ίδιο με το πριν

`static void sellcard(AgreementCards s , Character b;` Παιρνει τον χρηστη και την καρτα που εχει στο `inventory` του και την πουλα.

`static void dealcard()`; Ανοίγει ένα καινούργιο παραθυρό και παίρνει μια κάρτα συμφωνίας και την προβάλλει . Ο παίκτης έχει την δυνατότητα να την αγοράσει. Στην πραγματικότητα είναι η συνάρτηση που καλείται από τον `listener` του κουμπιού `deal`

`static void mailcard()`; Ανοίγει ένα καινούργιο παραθυρό και προβάλλει μια κάρτα μνημάτων . Έχει την δυνατότητα να καταλαβαίνει τι είδος κάρτας είναι με αποτέλεσμα να καταλαβαίνει τι είδος πράξη ο παίκτης πρέπει να κάνει. Είναι στην πραγματικότητα η συνάρτηση που καλείται από τον `listener` του κουμπιού `mail`.

`static void Sweep();` Είναι η συνάρτηση που καλείται όταν ο χρήστης πεσει σε `SweepStakes Tile` . Κάνει τις καταλληλες πράξεις για αυτό το ειδικό tile.

`static void lottery();` Είναι η συνάρτηση που καλείται αν ο χρήστης πεσει σε `Lottery Tile` . Ανοίγει ένα νέο παραθυρο και βάζει στους χρήστες να επιλεξουν διαφορετικούς αριθμούς.

`Static void radio();` Είναι η συνάρτηση που καλείται αν ο χρήστης πεσει σε `Radio Tile` .Ανοίγει νέο παραθυρο και δίνει στους χρήστες ζαρία να ρίξουν.

`Static void buyer();` Όταν ο χρήστης πεσει σε `Buyer Tile` καλείται αυτή η συνάρτηση και ο χρήστης μπορεί να πουλησει μια καρτα του.

`static void casino(Character c);` Συνάρτηση που καλείται αν πεσει σε `casino tile` και κάνει τις καταλληλες πράξεις.

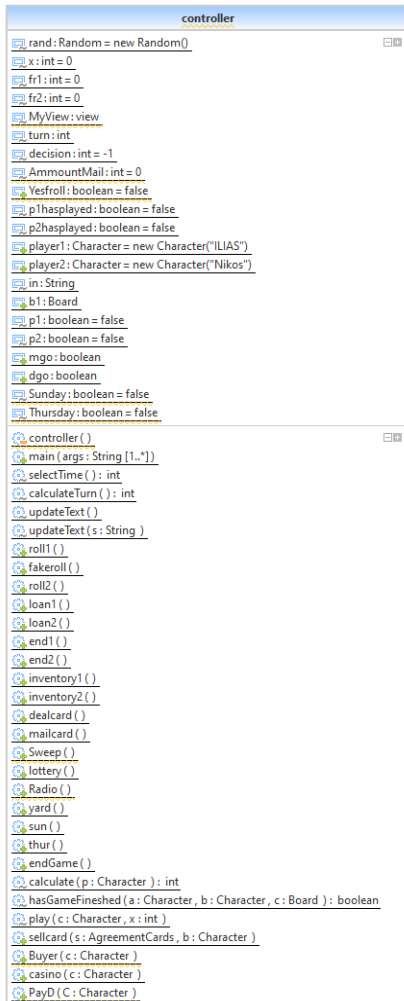
`static void yard()` Συνάρτηση που καλείται και αν ο παίκτης πεσει σε `YardSale Tile`

`Static void PayD()` Είναι η συνάρτηση που καλείται όταν ο χρήστης φτάσει στο τέλος του μηνά. Κάνει τις πράξεις που λενε οι οδηγίες.

`static void sun();` Είναι η συνάρτηση που καλείται όταν ο χρήστης πεσει σε `μερα Κυριακη` .Κάνει τις πράξεις που πρεπει (Συνηθως ο χρήστης πρεπει να το κάνει αφού πρωτα τελειωσει τις άλλες πράξεις του)

`static void thur()` το ίδιο με πριν απλα για τις μερες τις Πέμπτης.

`Static void endGame();` Τελειωνει το παιχνδι και βρισκει τον νικητη



Αυτό είναι το UML Class Diagramm της κλασης Controller .Μπορούμε να παρατηρήσουμε όλες τις ιδιαιτερες μεθοδους για κάθε Tile του παιχνιδιου αλλα και τα Attributes . Δυο πολύ σημαντικα είναι το board b1 = new Bord() και το view MyView = new view().

Package VIEW:

Το package view αναλαμβάνει τα γραφικά του παιχνιδιού και το interface με το οποίο οι Παικτες επικοινωνουν για να κανουν πραξεις στο παιχνιδι

Class VIEW:

Κλαση η οποια προβαλλει στην οθονη τα γραφικα του παιχνιδιου. Συγκεκριμενα αναπαριστα τα 31 Tiles του παιχνιδιου αλλα και το JackPot Tile με buttons που δεν εχουν actionListener. Επισης εχει 4 button ένα για κάθε χρηστη με τα οποια , οι παικτες μπορούν να αλληλεπιδρουν και να παιξουν το παιχνιδι. Τελος , εχει 3 JLabels που δειχνουν ενημερωτικα σχολια για το παιχνιδι.

Τα attributes:

JFrame Mp = new JFrame("PAYDAY"); Το Frame του παιχνιδιου

JPanel panel1 = new JPanel(); Το panel για την αναπαρασταση του παιχνιδιου

JPanel panel2 = new JPanel(); Το panel για την τοποθετηση κουμπιων για τους παικτες (στο μελλον θα χωριστεί σε 2 μικροτερα panels 1 για κάθε παικτη)

Jpanel panel3 = new JPanel(); Θα ήταν το 3ο panel αλλα δεν ακολουθησα αυτην αρχιτεκτονικη και λαθος μου καθως αναγκαστηκα να κανω το frame non resizable.

JTextArea info = new JTextArea(); Περιοχη που φαινονται ενημερωτικα σχολια για το παιχνιδι.

`public JLabel label1 = new JLabel("<html>Player 1
Money: 0
Loan: 0
bills: 0");` Label που δειχνει τα χρηματα του Player1

`public JLabel label2 = new JLabel("<html>Player 2
Money: 0
Loan: 0
bills: 0");` Label που δειχνει τα χρηματα του Player2

`public JLabel label3 = new JLabel("<html>JackPot Money:");` Label που δειχνει ποσα χρηματα εχουν μαζευτει στο jackpot Tile.

`public JButton roll1 = new JButton("Roll dice");` Το ζαρι του πρωτου παικτη

`public JButton roll2 = new JButton("Roll dice");` Το ζαρι του 2ου παικτη

`public JButton card1 = new JButton("My Deal Cards");` Το κουμπι που δινει στον παικτη την δυνατοτητα να ανοιξει το inventory του

`public JButton card2 = new JButton("My Deal Cards");` Ιδιο με το πριν απλα για άλλο παικτη

`public JButton loan1 = new JButton("Get a Loan");` ο κουμπι που δινει στον παικτη την δυνατοτητα να παρει δανειο

`public JButton loan2 = new JButton("Get a Loan");` Το ίδιο αλλά για τον άλλο παίκτη

`public JButton turn1 = new JButton("End Turn");` Το πατάει ο παίκτης για να τελειώσει την σειρά του και να αρχίσει η σειρά του άλλου παίκτη

`public JButton turn2 = new JButton("End Turn");` Το ίδιο με πριν απλά για τον άλλο παίκτη

`public JButton dice1 = new JButton();` Αναπαράστα γραφικά το ζαρι του πρώτου παίκτη

`public JButton dice2 = new JButton();` Αναπαράστα γραφικά το ζαρι του δεύτερου παίκτη

`public JButton mail = new JButton();` Το κουμπί που επιτρέπει στον χρήστη να πάρει κάρτα μνηματος

`public JButton deal= new JButton();` Το κουμπί που επιτρέπει στον χρήστη να πάρει κάρτα συμφωνίας

Οι μέθοδοι:

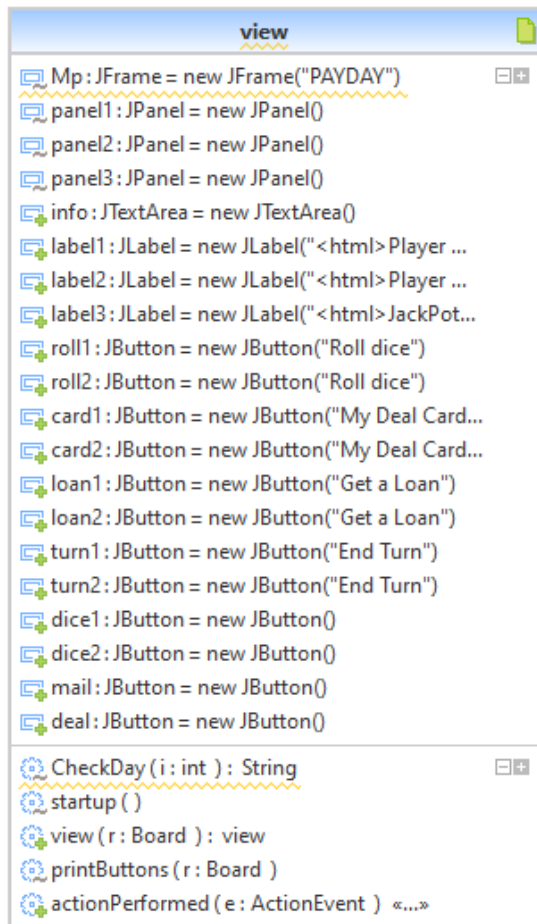
`public view(Board r)` Δημιουργεί την ένα instance της κλάσης view που

Αναλαμβάνει την δημιουργία παραθύρου για προβολή γραφικών

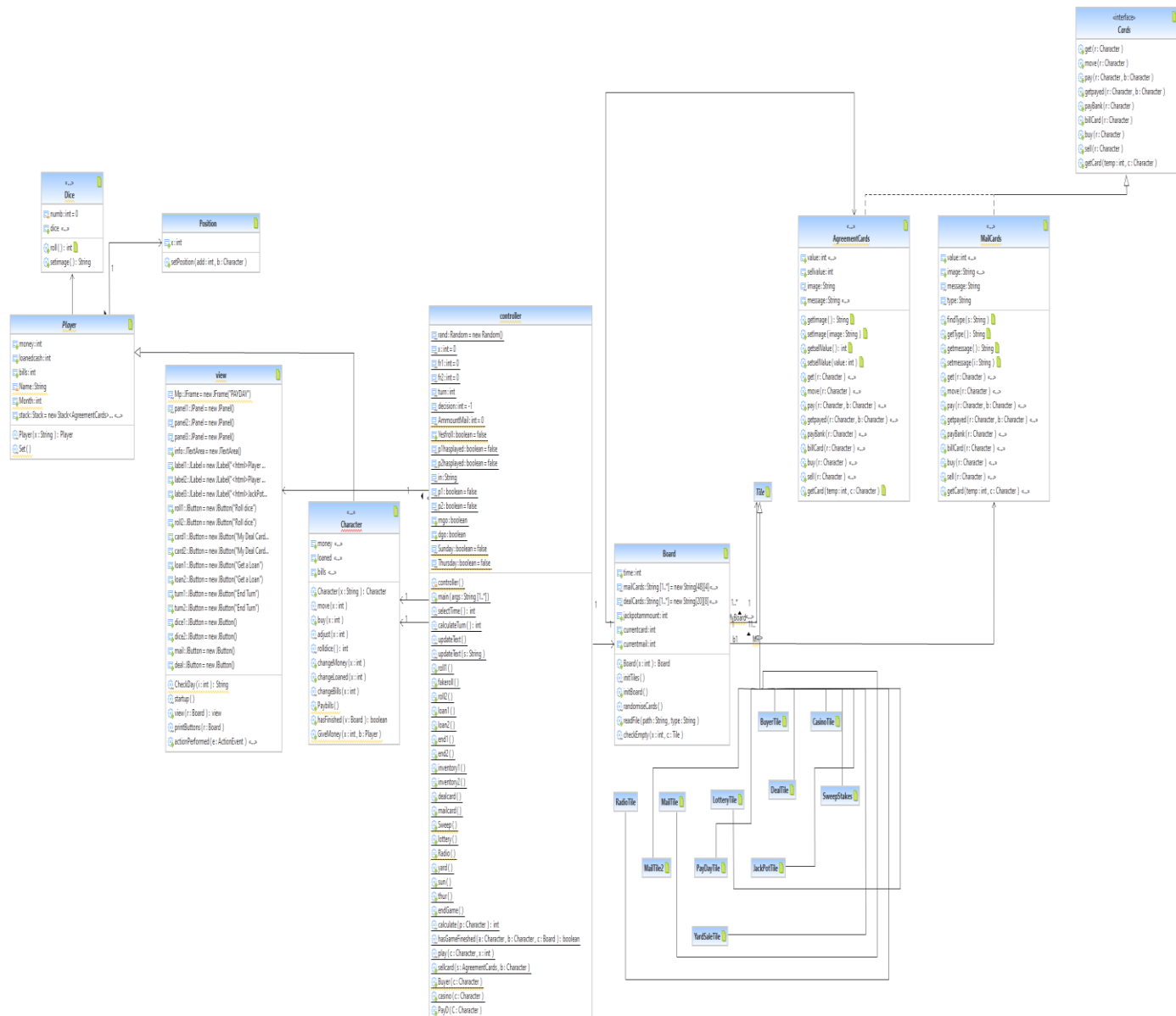
`void printButtons (Board r)` Τυπώνει όλα τα πλακίδια του board

`String CheckDay(int i)` Τυπώνει τις μέρες του κάθε πλακιδίου που ανήκει στο board

void startup(); Πολύ σημαντική δίνει , λειτουργικότητα στα κουμπια.



UML PROJECT DIAGRAMM



Από το UML CLASS DIAGRAMM μπορούμε να μελετήσουμε τις διασυνδεσεις μεταξύ των κλάσεων. Αρχικά , η κλάση Player συνδεεται με τις κλάσεις Dice (η οποία δίνει δυνατοτητα στον παικτη να ρίξει ζαρι) και Position (η οποία αποθηκευει την θέση του παικτη στο board).Μετα , η κλάση Character συνδεεται με την player αφού την κανει extend και παίρνει τα attributes και προσφέρει νέες μεθοδους.Στην συνέχεια , οι κλάσεις Agreement Cards , DealCards οι οποίες κανουν extend Cards και αντιπροσωπευουν τις καρτες του παιχνιδιου.Παραλληλα, το Board συνδεεται με τις καρτες και την κλάση Tiles η οποία αποτελείται από πολλά instances της.Η κλάση controller από το package controller τωρα συνδεεται με τη Board και την Character επειδη ετσι το controller εχει συνδεση με το package model .Επισης συνδεεται με την κλάση View από το package view που εχει αναλαβει τα γραφικα του παιχνιδιου.Ετσι βλέπουμε ότι κάθε package εχει καποια σχετικη ανεξαρτησια καθώς ο μονος τροπος για να επικοινωνισουν view και model είναι μεσω του εγκεγαλου του παιχνιδιου δηλαδη του controller.

Λειτουργικότητα

Το παιχνίδι αρχικά αποφασίζει πια θα είναι η σειρά των παικτών (μία διαδικασία η οποία γίνεται τυχαία κάθε φορά) και αρχικοποιεί τα tiles του board και τις κάρτες του παιχνιδιού(τα οποία επίσης κάθε φορά επιλέγονται σε τυχαία σειρά.) Στην συνέχεια δίνεται η δυνατότητα στους παίκτες να παίζουν με εναλλαξ σειρά . Κάθε φορά που κάποιος παίκτης πεφτει σε ειδικό Tile βγαίνει ενημερωτικό σχολίο που του λέει τι να κάνει καθώς και σε ποια θέση βρίσκεται .Σε περίπτωση που ένας παίκτης έχει τελειώσει ο άλλος παίκτης συνεχίζει το παιχνίδι μέχρι να τελειώσει . Οι κάρτες μόλις τις χρησιμοποιήσει κάποιος απορρίπτονται εκτός αν τελειώσουν οι κάρτες που στην συγκεκριμένη περίπτωση ο ξαναμπαινουν πίσω. Αν κάποιος χρήστης θέλει να αγοράσει κάτι και δεν του φτάνουν τα λεφτά ο κωδικας αυτοματα του βάζει δανεικά λεφτά.Παραλληλα οι παίκτες μπορούν να πουλήσουν κάρτες συμφωνίας μονο αν πεσουν σε buyer Tile ενώ σε περίπτωση που η κάρτα μνηματος είναι move to buyer αν ο παίκτης βρίσκεται σε θέση που δεν υπάρχει άλλος buyer μέχρι το τέλος του μηνά μνει στην θέση του.Το tile Payday δίνει 3500\$ στον παίκτη , του επιβάλλει φορο , πληρωνει όλα τα bills ανεξαρτητα αν δεν έχει αρκετά καθαρά χρηματα και του δίνει δυνατότητα να ξεπληρωσει ένα ποσο του δανειου .Αν είναι τελευταίος μηνας ο παίκτης ξαναριχνει το ζαρι για τελευταία φορά ώστε να <<ξεπεράσει >> την μερα PayDay.

Σημειωση

Το μονο functionality του παιχνιδιου που είναι λιγο buggy είναι οι μερες Κυριακη και παρασκευη .Ενω στον κωδικα το εχω βαλει να γινονται μετα από τις κανονικες πραξεις του κάθε Tile το παραθυρο βγαίνει ταυτοχρονα .Δεν περιμενω να παρω ολους τους βαθμους από τις μερες Κυριακη Πempπη αλλα όχι και κανεναν. Όταν ο χρηστης πεσει σε μερα οπου πρεπει να σηκωσει mail η deal card προτιμοτερο είναι να παιξει πρωτα thurdsay η sunday . Σε buyer tile και lottery δεν έχει σημασια ουτε στο casino. Προβλημα παρουσιαζεται σε radio Tile η yardsale- sweepstake οπου

μπαιδευονται οι σειρες. Σας ευχαριστω για την υπομονη εκ των προτετων
😊.

Τελος