

Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών
ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός
Διδάσκων: Ι. Τζιτζικας
Χειμερινό Εξάμηνο 2019-2020

SORRY! (A BOARD GAME)

ΟΝΟΜΑ : Μερκούρης Λουκάς

A.M. : 3131

Περιεχόμενα

1. Εισαγωγή.....
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

ΕΙΣΑΓΩΓΗ

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller). Έτσι, σκοπός μας είναι ο Controller να είναι ο συνδετικός κρίκος των Model και view. Οπότε στη συνέχεια της αναφοράς μας θα αναλύσουμε λίγο ιδιαίτερα τα κομμάτια του Model και Controller που είναι σημαντικά για αυτή τη φάση και τέλος θα αναφερθούμε και λίγο στο view.

Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Abstract class card:

Εδώ θα φτιάξω την αφηρημένη κλάση card η οποία μοντελοποιεί μια κάρτα του παιχνιδιού και ορίζει τα γνωρίσματα της και τις χρήσιμες μεθόδους της.

Η αφηρημένη κλάση αυτή έχει τις μεθόδους

1. Public abstract boolean isActive(); (observer)
returns true if card is not played, else return false.
2. Public void setActive(boolean isActive);
Transformer (Mutative), sets the isActive attribute.
3. Public string toString();
Returns a String Representation of a Card.
4. Public void movePawn(Pawn pawn);Transformer
(Multative) checks if the pawn can move, if so then moves the pawn.

Class NumberCard extends Card

Οι μέθοδοι και τα attributes είναι :

1. Private int value; // the value of the card.
2. Private int numCardPlayed; // how many times the card has been played.
3. Public String toString(); //returns "NumberCard".
4. Public void movePawn(); //moves the pawn.
1. Public void getValue(); accessor (Selector)
gets the value of the card
2. Public int setValue(int value); Transformer(Multative)
sets the value of the card.
3. Public void getNumCardPlayed (); accessor (Selector)
gets the value of the card
4. Public int setNumCardPlayed (int numCardPlayed);
Transformer(Multative) sets the value of the card.

Class SorryCard extends Card

Οι μέθοδοι και τα attributes είναι :

1. Public boolean getPawn(Pawn pawn); Accessor(Selector)
,Gets a pawn and checks if it can be played, returns
canBePlayed.
2. Private boolean canBePlayed; true if pawn can be played
, else canBePlayed is false .
3. Public Pawn movePawn(Pawn pawn); moves the pawn if
canBePlayed is true ;
5. Public String toString(); //returns "SorryCard".

Class SimpleNumberCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Public SimpleNumberCard(); constractor for the simple card.

Class NumberOneCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Public boolean setStart(Pawn pawn); Transformer(Multative)
A pawn can start.
2. Public NumberOneCard();

Class NumberTwoCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Public int setTurn(); Transformer(Multative) sets the
turn to be the same as before
2. Public boolean setStart(Pawn pawn); Transformer(Multative)
A pawn can start.
3. Public NumberTwoCard();

Class NumberFourCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Public NumberFourCard() // sets the value of the card to
-4 (to move backwards).

Class NumberSevenCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Private boolean moveBoth;
2. Private int move1,move2;
3. Public void getMoveBoth(); accessor (Selector)
true if player wants to play both pawns.
4. Public boolean setMoveBoth (boolean moveBoth);
Transformer(Multative) sets the moveBoth
variable .
5. Public void getMove1(); accessor (Selector)
gets the move1.
6. Public int setMove1(int move1); Transformer(Multative)
sets move1
7. Public void getMove2(); accessor (Selector)
gets the move2.
8. Public int setMove2(int move2); Transformer(Multative)
sets move2
9. Public boolean canBePlayed(); throws
ilegallargumentexpreciton // return true if move1 + move2 = 7
false otherwise .
10. Public NumberSeverCard();

Class NumberTenCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Public boolean canPlayForward;
2. Public int backORfront; //0 to go forward 10 , 1 to go back 1
3. Public void getCanPlayForward(); accessor(Selector) gets the canPlayForward.
4. Public boolean setCanPlayForward(boolean canPlayForward); Transformer(Multative) sets the canPlayForward.
5. Public void get backORfront (); accessor(Selector) gets the backORfront.
6. Public int setCanPlayForward(int backORfront); Transformer(Multative) sets the backORfront.
7. Public NumberTenCard(); constractor

Class NumberElevenCard extends SimpleCard

Οι μέθοδοι και τα attributes είναι :

1. Private int move;
2. Public void getMove (); accessor(Selector) gets the backORfront.
3. Public int setMove (int move); Transformer(Multative) sets the backORfront.

4. Public boolean checkStart(Pawn pawn); //checks if the pawn is in start area.
5. Public boolean checkSafe(Pawn pawn); // checks if the pawn is in safe area.
6. Public NumberElevenCard();

In order checkStart and checkSafe to initiate move has to be 1.

Class Square

Εδώ μοντελοποιώ το κάθε τετράγωνο του παιχνιδιού, το οποίο έχει μια συγκεκριμένη θέση στο ταμπλό και ένα συγκεκριμένο χρώμα ,αν έχει πιόνι πάνω του ή αν είναι άδειο προς το παρόν . Επίσης θα ορίσω και τις υποκλάσεις του (home square , slide square κτλπ).

Οι μέθοδοι και τα attributes είναι :

1. Private int number;
2. Private boolean hasPawn;
3. Private int squareValue; // sets the value of the square , 1 for simple square , 2 start square , 3 slide square , 4 safety zone square , 5 home square
4. Public void getNumber(); accessor (Selector)
gets the number of the block
5. Public int setNumber(int number); Transformer(Multative)
sets the number of the block

6. `Public int getPawn();` accessor (Selector)
gets the value if it has pawn.
7. `Public int setPawn(boolean hasPawn);` Transformer(Multative)
sets the block to has a pawn .
8. `Public int getSquareValue ();` accessor (Selector)
gets the value of the square.
9. `Public int setSquareValue (boolean hasPawn);`
`Transformer(Multative)` sets the value of the square .
10. `Public Square(int num, boolean hasPawn);` constractor

Class StartSquare extends Square

Οι μέθοδοι και τα attributes είναι :

1. `Private int activepawns;` // how many pawns are active in the start square.
2. `Public int getActivepawns();` accessor(Selector)
returns active pawns in start square.
3. `Public void setActivepawns();` Transformer(Multative)
sets the active pawns in start square .
4. `Public StartSquare();`

Class SimpleSquare extends Square

Οι μέθοδοι και τα attributes είναι :

1. Public SimpleSquare();

Class SlideSquare extends Square

Εδώ είναι η τσουλήθρα . Άρα θα πρέπει να ορίσω αν είναι τσουλήθρα τεσσάρων ή πέντε τετραγώνων.

Οι μέθοδοι και τα attributes είναι :

1. Private int squares;
2. Public void getSquares(); accessor(Selector)
gets the number of squares in the slide .
3. Public int setSquares(int squares); Transformer(Multative)
sets the number of squares in the slide .
4. Public SlideSquare();

Class StartSlideSquare extends SlideSquare

Οι μέθοδοι και τα attributes είναι :

1. Private boolean checkColor; //checks the color of the pawn.
2. Public void getCheckColor(); accessor(Selector) // gets the
color of the pawn at the start of the slide.
3. Public boolean setCheckColor(boolean checkColor);
Transformer(Multative) sets the color of the pawn at the start
of the slide.
4. Public StartSlideSquare();

Class InternalSlideSquare extends SlideSquare

Οι μέθοδοι και τα attributes είναι :

1. Private boolean checkColor;
2. Public void getCheckColor(); accessor(Selector) gets the color of the pawn at the start of the slide.
3. Public boolean setCheckColor(boolean checkColor); Transformer(Multative) sets the color of the pawn at the start of the slide.
4. Public InternalSlideSquare();

Class EndSlideSquare extends SlideSquare

Οι μέθοδοι και τα attributes είναι :

1. Private boolean checkSameColor;
2. Private boolean checkForwardMove;
5. Public void getCheckSameColor (); accessor(Selector) // gets the color of the pawn at the end of the slide.
6. Public boolean setCheckSameColor(boolean checkSameColor); Transformer(Multative) sets the color of the pawn at the start of the slide.
7. Public void getCheckForwardMove (); accessor(Selector) gets the color of the pawn at the start of the slide.
8. Public boolean setCheckForwardMove (boolean checkForwardMove); Transformer(Multative) // sets the color of the pawn at the start of the slide.

Class SafetyZoneSquare extends Square

Οι μέθοδοι και τα attributes είναι :

1. Private Color color; // color of safety zone square.

2. Public boolean setProtect(Pawn pawn); // sets a pawn to be protected.
3. Public boolean checkFrontMove(); // checks if a pawn enters by moving forward.
4. Public SafetyZoneSquare();

Class HomeSquare extends Square

Οι μέθοδοι και τα attributes είναι :

1. Private int howManyPawns; // how many pawns home has.
2. Public boolean setLock(Pawn pawn); Transformer(Multative) locks a pawn so it cant move, true pawn is lock , false pawn is unlocked.
3. Public boolean checkSquaresLeft(Pawn pawn); Observer checks how many squares left till home

Class Player

Σε αυτή την κλάση θα ορίσουμε έναν παίχτη μέσα στο παιχνίδι .

Τα attributes είναι :

1. Private String name; // the name of a player.
2. Private boolean turn;// the turn of the player.
3. Private Color color; // the color of a player.
4. Private int choice; // the choise of a player (fold or move a pawn).
5. Private int id; // the id of a player.

Οι μεθοδοι είναι :

6. Public void getTurn(); accessor (Selector) gets the turn of the player.
7. Public boolean setTurn(boolean turn); accessor (Selector), sets the turn of the player.
8. Public void initPlayer(); Transformer(Multative) initialize a player
9. Public int getId(); accessor (Selector) return the id of a player.
10. Public void setId(); Transformer(Multative) return the id of a player.
11. Public int getName(); accessor (Selector) return the name of a player.
12. Public void setName(); Transformer(Multative) return the name of a player.

13. Public Color getColor(); accessor (Selector) gets
the color of the player;
14. Public void setColor(); Transformer(Multative) sets
the color of the player;
15. Public int getChoise(); accessor (Selector) gets
the choise of the player.
16. Public void setChoise(); Transformer (Multative) sets the
choise of the player.
17. Public boolean hasPlayed(); Observer
returns if a player has played his round.

Class Pawn

Εδώ δημιουργώ την κλάση pawn η οποία είναι υπευθινή για την δημιουργία ενός πιονιού, το οποίο έχει ένα συγκεκριμένο χρώμα ,βρίσκεται σε ένα συγκεκριμένο τετράγωνο και είναι είτε ενεργο είτε ανενεργο ή να έχει φτάσει στον τελεικο του προορισμο (home square).

Οι μέθοδοι και τα attributes είναι :

1. Private Color color; // the color of a pawn
2. Private int block; // the block that the pawn is .
3. Private boolean is active; // the activity of a pawn (true is active , false is not active).
4. Private boolean isLocked;
5. Private boolean isprotected;
6. Public void get isLocked (); accessor (Selector) gets the locked filed of the pawn.
7. Public Color setColor (boolean isLocked); Transformer (Multative) sets the lock of the pawn.
8. Public void getProtected (); accessor (Selector) gets the protect field of the pawn.
9. Public Color setProtected (boolean protected); Transformer (Multative) sets the protected field of the pawn.
10. Public void getColor(); accessor (Selector) gets the color of the pawn.
11. Public Color setColor(Color color); Transformer (Multative) sets the color of the pawn.
12. Public void getBlock(); accessor (Selector) gets the block of the pawn.
13. Public int setBlock(); Transformer (Multative) sets the block of the pawn.
14. Public void getIsActive(); accessor (Selector) gets if the pawn is active or not .
15. Public boolean setIsActive(boolean is active); Transformer (Multative) sets if the pawn is active or not .

Class Deck

Εδώ δημιουργώ το ταμπλό , τις κάρτες και τα πιόνια για κάθε παίχτη ,επίσης φτιαχνω μεθόδους για την μετακίνηση ενός πιονιου.

Οι μέθοδοι και τα attributes είναι :

1. Private ArrayList<SimpleSquare> simplesquare // Array list that saves all the simple squares of the deck.
2. Private ArrayList<StartSquare> startsquare // Array list that saves all the start squares .
3. Private ArrayList<Square> homesquare // Array list that saves all the home squares.
4. Private ArrayList<SafetyZoneSquare> safetysquares // Array list that saves all the safety squares;
5. Private ArrayList<Card> cards // Array list that saves all the cards of the game .
6. Public Card initCards(); Transformer(Multative) initialize the cards .
7. Public Deck initSquares(); Transformer(Multative) initialize pawns and squares.
8. Public void chageBlock(Pawn pawn, Square oldsquare,Square newSquare, int oldPos, int newPos); // takes a pawn and moves it to the new square.
9. Public checkIfThereIsPawn(Pawn pawn , int squarepos) // checks if the given squarepos has a pawn.
10. Public boolean movePawn(Pawn pawn, Card card) //Moves the pawn by the value of a card , returns true if the pawn moved succesfull or else return false .

11. Public boolean isCardsEmpty(Card[] cards); Observer
returns true if the card deck is empty , false else .
12. Public Card shuffleCards(Card[] cards);
Transformer(Multative)
13. Public ArrayList<Card> getCards() // gets the array list that
contains the cards.
14. Public void setCards(ArrayList<Card> cards) // sets the
array list with the given cards.
15. Public Card getLastCardPlayed () // returns the card that is
about to be played.
16. Public void setLastCardPlayed() // sets the card that is
about to be played.
17. Public ArrayList<StartSquare> getStartSquare() // returns
the list of the start squares.
18. Public void setStartSquare(ArrayList<StartSquare>
startsquare) // sets the start square array list .
19. Public void movePawn(Pawn pawn);
Transformer(Multative)
20. Public boolean canFold(Player player); Observer
return true if player can fold.
21. Public Card getCard(); accessor(Selector)
gives player a card.

Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Εδώ θα σχεδιάσω τον Controller το “μυαλό” του παιχνιδιού. Είναι υπεύθυνος για την δημιουργία παιχνιδιών , παιχτών , πιονιών , είναι αυτός που θα συνδέσει το πακέτο View με το Model . Στην ουσία θα παίρνει τις επιλογές του χρήστη μέσω των γραφικών και θα πραγματοποιεί τις κατάλληλες ενέργειες στο παιχνίδι. Προφανώς είναι και υπεύθυνος για τον τερματισμό του παιχνιδιού.

Class Controller

Τα attributes είναι :

1. Private int turn;
2. Private Player p1,p2;
3. Deck deck;
4. View view;

Οι μέθοδοι είναι :

1. Public int setThisTurn() Accesors returns a random number between 0 and 1 for the turn of which player
2. Public int seeTurn(); Accesors(Selector) return the turn of which player.
3. Public int setTurn(); Transformer (Multative) sets random turn in the start of the game .
4. Public boolean checkMove(Pawn pawn); accessor (Selector) returns true if the move is valid.
5. public boolean not_started(); Observer Return true if the game has not started false otherwise
6. Public boolean checkIfGameIsOver(); Observer checks if the game is over.

7. `public void initialize() ; Transformer (Mutative)`
initialize Deck sets the round, create players,create the view .
8. `Public void update() //` sets the cards to be played, ran every array list and update them(which square has which pawn).
9. `Public void removeAndReplaceCard() //` remove a closed card and update the cur card.
10. `Public void setListeners() //` sets the listeners for the buttons
11. `Private class CurrCardListener implements`
`MouseListener //` listener to current card.
12. `Private class foldListener implements`
`MouseListener //` listener to fold button.
13. `Private class CardListener implements`
`MouseListener //` listener to closed card.
- 14.

Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Αυτό το πακέτο θα αποτελείτε από μια κλάση. Η κλάση αυτή είναι υπεύθυνη για όλο το οπτικό κομμάτι του παιχνιδιού. Αρχικά θα δημιουργεί ένα frame και μέσα σε αυτό θα υπάρχει ένα JLayerdPane στο οποίο θα υπάρχει ένα jtextarea για να κρατάμε πληροφορία για το ποιος παίχτης έχει σειρά , πόσες κάρτες έχουν μείνει, ένα JButton για το fold 2 ακόμα JButtons για τις κάρτες(ανοιχτές και κλειστές) καθώς και 2 JLabels για τι ποια κάρτα είναι ποια .Έπειτα στα αριστερά του frame , θα δημιουργήσουμε 64 + 10 τετράγωνα

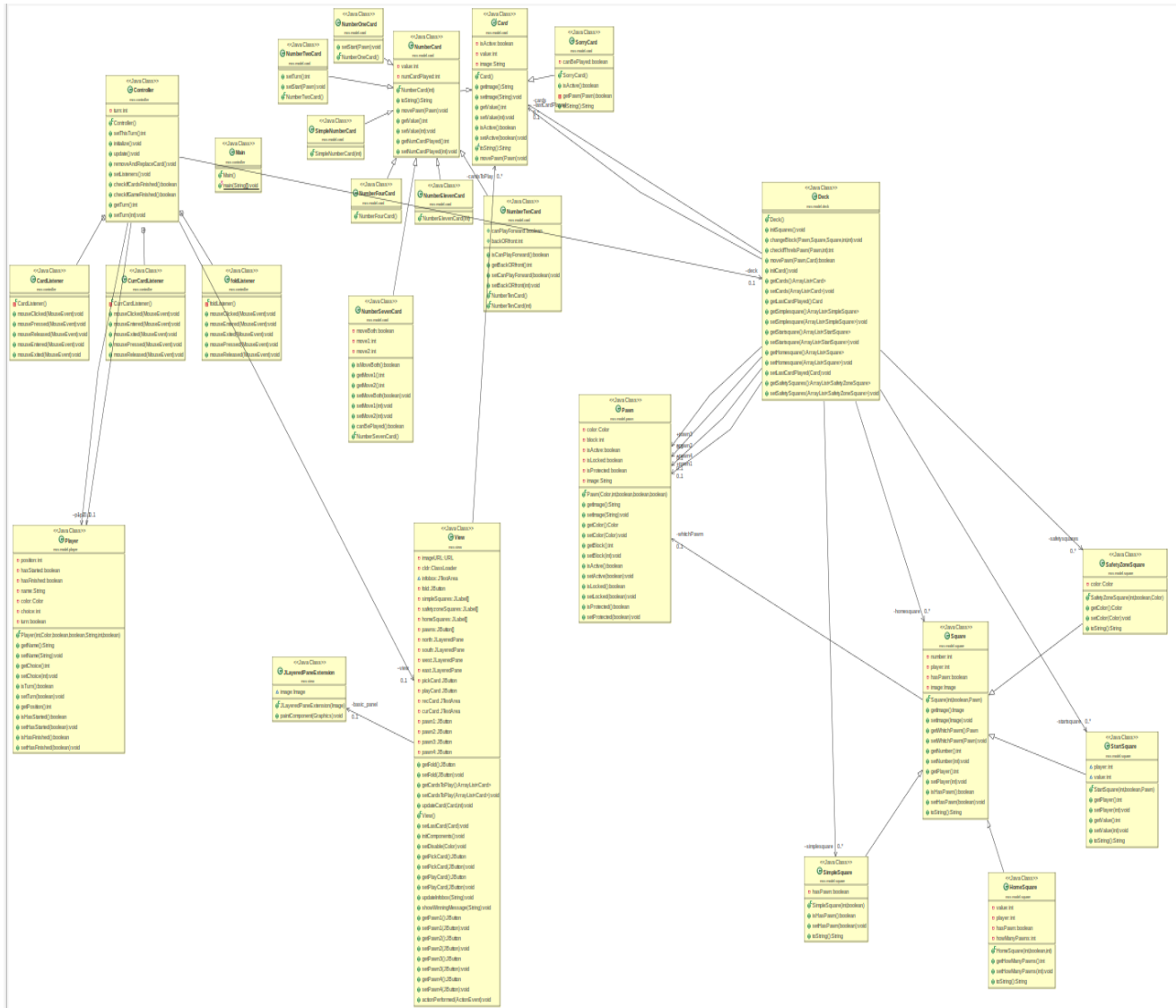
(JLabel)εκ των οποίων τα 9 θα είναι τσουλήθρες για το ένα χρώμα και άλλα 9 για το άλλο, 10 για τα safe areas του κάθε παίχτη και τα υπόλοιπα απλά τετράγωνα .Θέλουμε ακόμα 2 JLabel+text για το start του κάθε παίχτη , και 2 αλλά 2 για το Home του κάθε παίχτη με 2 κουμπιά για τα πιόνια .

Τα attributes και οι μεθοδοι είναι :

1. Private JLayeredPaneExtension basic_panel; // the basic panel.
2. Private ClassLoader cldr;
3. Private JButton fold;
4. JTextArea infobox;
5. JLabel[] simplesquares; //jlabels for simple squares.
6. JButton[] pawns ; // buttons for the pawns.
7. JButton pickCard,playCard; // buttons for the closed card and current card.
8. Private JTextArea recCard , curCard // simple text for the buttons info .
9. Public JButton getFold() // gets the fold button

10. Public void setFold(JButton fold) // sets the fold button.
11. Public void updateCard(Card c , int position) // Transformer , updates the cards.
12. Public View() // Constructor of the class view.
13. Public void setLastCard(Card c) // sets the last card.
14. Public void initComponents() throws IOException //initialize all the buttons labels and texts and adds them on basic_panel.
15. Public void updateInfobox() // update the info box.
16. Public void showWinningMessage() // show a message to basic_panel which player won.
17. Getters and setters for buttons and pawns.
18. Public void actionPerformed() // do an anction when the card is pressed

Τελικό Διάγραμμα UML:



ΤΕΛΟΣ