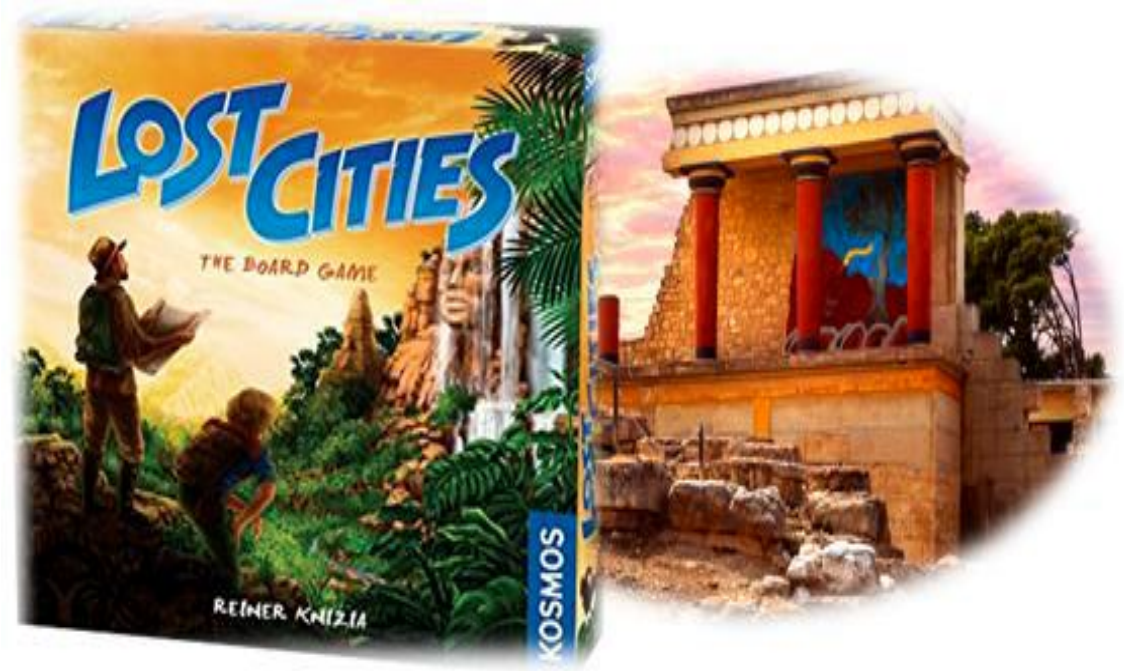
OBJECT-ORIENTED PROGRAMMING HY-252
PROJECT 2017-2018



Ioannis Manioros

ΠΕΡΙΕΧΟΜΕΝΑ

Η υλοποίηση της εργασίας θα βασίζεται πάνω στο μοντέλο MVC (Model View Controller).Έτσι , ο Controller να είναι ο συνδετικός κρίκος των Model και view.

Package Model

Σε αυτό το πακέτο θα περιέχονται διεπαφή Card,οι κλάσεις NumberCard,
SpecialCard, Minotaur και Ariadne. Οι κλάσεις που κληρονομούν την Special Card είναι οι Minotaur και Ariadne. Επιπλέον σε αύτο το πακετο υπαρχουν τα εξής :

- Package finding το οποίο περιέχει τα εξής:
  - Αφηρημένη κλάση Finding
  - Κλάση Fresco
  - Κλάση RareFinding
  - Κλάση SnakeGoddess
- Package collection το οποίο περιέχει τα εξής:
  - Αφηρημένη κλάση Collection
  - Κλάση FrescoCollection
  - Κλάση RareCollection
  - Κλάση SnakeGoddessCollection
- Package Sullogi το οποίο περιέχει τα εξής :
  - Κλάση Sullogi
- Package City το οποίο περιέχει τα εξής :
  - Κλάση City
- Package Board το οποίο περιέχει τα εξής :
  - Κλάση Board
  - Κλάση Path’
  - Κλάση Pawn
- Package player :
  - Κλάση player
- Package position :
  - Κλάση Position
  - Κλάση FindingPosition
  - Κλάση SimplePosition

Interface Card and Card’s methods

Με τη διεπαφή Card μπορούμε να δημιουργήσουμε αντικείμενα τύπου Card χωρίς να έχουμε προσδιορισεί αν είναι NumberCard ή SpecialCard.

Το interface αυτό μας παρέχει την μεθόδο :

- public City getCity(); Accessor
  Returns the city of Card

Στη συνέχεια έχουμε την NumberCard και την SpecialCard που υλοποιούν την Card.

Class NumberCard

Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η

κλάση αυτή .

Τα attributes:

- private int value; //The value of a number Card.
- private City city; //The City of a number Card.

Ο ι μέθοδοι :

- public NumberCard(int value, City city); Constructor

  Creates a new number Card

- public void setValue(int value); Transformer

  sets the card's value

- public int getValue(); Accessor

  returns the card's value

- public City getCity(); accessor

  Returns card's city

- public String toString();

  Returns the string representation of a number card

Class SpecialCard

Εδώ θα αναφέρουμε τα attributes και τις μεθόδους που έχει η

κλάση αυτή .

Τα attributes:

- private City city; // The City of a number Card.

Ο ι μέθοδο ι :

- public SpecialCard(City city); Constructor

  Creates a new number Card

- public City getCity(); accessor

  Returns card's city

- public String toString();

  Returns the string representation of a special card

Οι κλάσεις Minotaur και Ariadne είναι υπο-κλάσεις της κλάσης SpecialCard.

Class Minotaur

Ο ι μέθοδο ι :

- public Minotaur(City c); Constructor

  Creates a new Minotaur Card

- public String toString()

  Returns the string representation of a Minotaur card

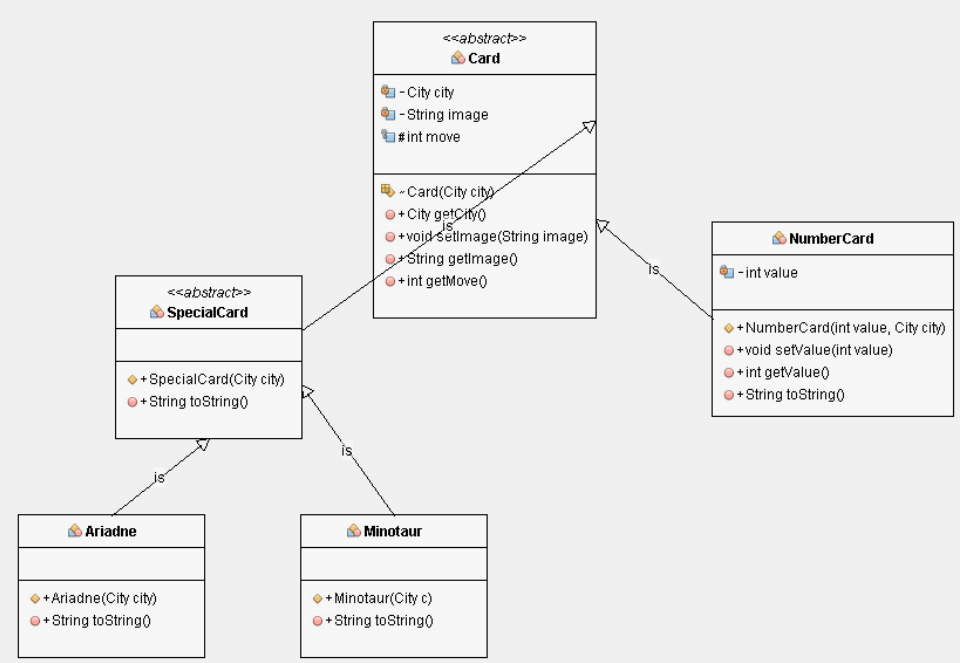
Class Ariadne

- public Ariadne(City city); // Constructor

  Creates a new Ariadne Card

- public String toString()

  Returns the string representation of a Ariadne card



Package finding

Σε αυτο το πακέτο υλοποιούμε τα αρχαιλογικά ευρήματα.

Abstract class finding

Abstract class finding που υλοποιεί τα αρχαιολογικά ευρήματα.

Οι μεθόδοι της Abstract class finding :

- public int getPoints(); // Accessor

  returns the Finding's points
- public void setPoints(int points); //Transformer

  sets the Finding's points
- public String toString()

  Returns the string representation of Finding

Class Fresco

Η κλάση Fresco κάνει extend την finding. Η κλάση Fresco αναφέρετει στις τοιχογραφίες που αποτελούν αρχαιολογικό εύρημα.

Τα attributes:

private int points; //Fresco 's points

private String name; //Fresco 's name

private final boolean collectable; // collectable = false δηλαδή δεν μπορείς να το συλλέξεις.

Οι μεθόδοι της κλάσης Fresco:

- public Fresco(); // Constructor

  Creates a new Fresco

- public int getPoints(); //Accessor

  returns the Fresco's points

- public void setPoints(int points); //transformer

  sets Fresco's points

- public String getName(); //Accessor

  returns the Fresco's name

- public void setName(String name); // transformer

  sets Fresco's name

Class RareFinding

Η κλάση Class RareFinding κάνει extend την finding.Η κλάση RareFinding αναφέρετει στα σπάνια ευρήματα που είναι γενικώς αρχαιολογικό εύρημα.

Τα attributes:

private int points; //RareFinding 's points

private String name; //RareFinding 's name

private final boolean collectable = true; // rare finding can been collected

Οι μεθόδοι της κλάσης RareFinding :

- public RareFinding(int points, String name); //Constructor

  Creates a new RareFinding

- public int getPoints(); //Accessor

  returns the RareFinding's points

- public void setPoints(int points); //transformer

  sets RareFinding's points

- public String getName(); //Accessor

  returns the RareFinding's name

- public void setName(String name); //transformer

  sets RareFinding's name

Class SnakeGoddess

Η κλάση SnakeGoddess κάνει extend την finding. Η κλάση SnakeGoddess αναφέρετει στα αγαλματάκια της Θεας των φιδιών που είναι γενικώς αρχαιολογικό εύρημα.

Τα attributes:

private int points;//SnakeGoddess's points

private final boolean collectable = true;// Snake goddess can been collected

Οι μεθόδοι της κλάσης SnakeGoddess:

- public SnakeGoddess(); //Constructor

  Creates a new SnakeGoddess

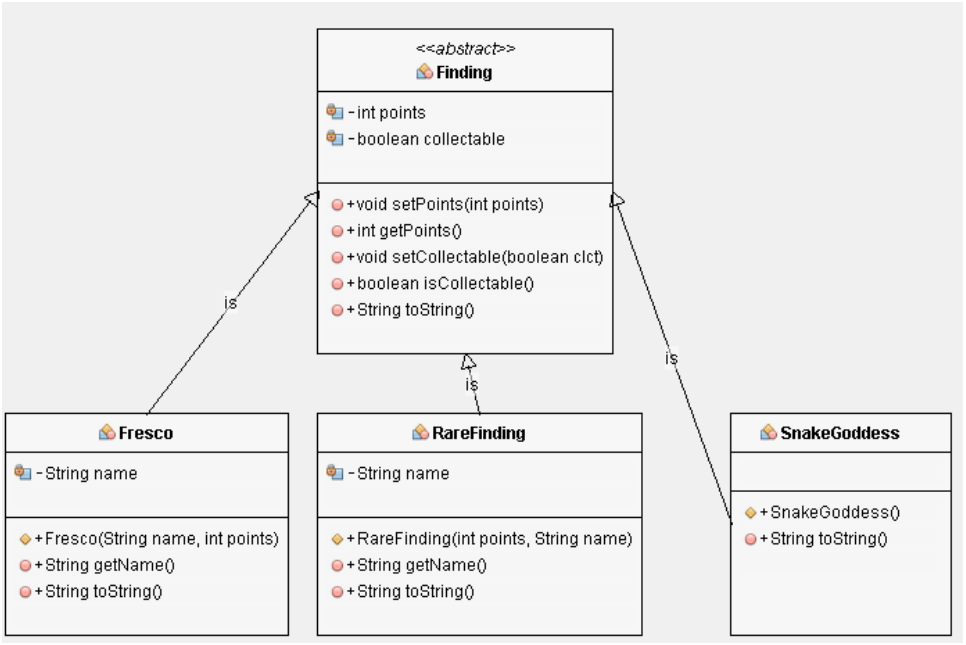
- public int getPoints(); //Accessor

  returns the SnakeGoddess's points

- public void setPoints(int points); //transformer

  sets SnakeGoddess's points

uml finding



Package Sullogi

Η κλάση Sullogi υλοποιεί μια συλλογή από κάρτες.

Class Sullogi

Τα attributes:

private ArrayList<Card> cards; //a Card ArrayList

Οι μεθόδοι της κλάσης Sullogi:

- public Sullogi(); //Constructor

  Creates a new Sullogi with a new card ArrayList.

- public void initCards(); //Transformer

  Initializes and shuffles cards.

- public boolean isEmpty(); //Observer

  Returns true if this list contains no elements.

- public void addCard(Card i); //Transformer

  Adds a card to the list.

- public void removeCard(Card i); //Transformer

Removes a card from the list.

- public int size(); //Transformer

  Returns the size of a list.

- public Card getCard(int index); //Accessor

  returns the card in position 'index'

- public void clearAll(); //Transformer

  Clears an ArrayList

- public ArrayList<Card> getCards(); //Accessor

  returns all the cards

- public Card drawCard(); // Accessor

  return a card

Package City

Class City

Είναι μια enum class που περιέχει τις πόλεις (Knossos,Malia,Faistos,Zakros)

Package Board

Class Board

Τα attributes:

Path[] path;; //one path for each city

Οι μεθόδοι της κλάσης Board:

public Board(); //Constructor

Creates a new Board

Class Path

Η κλάση Path υλοποιεί ένα μονοπάτι.

Τα attributes:

private City anaktoro; //path's city

private ArrayList<Position> path = new ArrayList<>(9); //path's 9 positions

private final int checkPoint = 7; // check point in the path

Οι μεθόδοι της κλάσης Path:

- public Path(City anaktoro); //Constructor

  Creates a new path with "city" City.

- public void setAnaktoro(City anaktoro); //Transformer

  sets the Anaktoro's city

- public void setPath(ArrayList<Position> path); //Transformer

  sets path

- public City getAnaktoro(); //Accessor

  returns the path's city

- public ArrayList<Position> getPath(); //Accessor

  returns the position in path

- public int getCheckPoint(); //Accessor

  returns the check point

- Position getPosition(int thesi); //Accessor

  returns the position

Class Pawn

Η κλάση Pawn υλοποιεί ένα πιόνι.

Τα attributes:

Path anaktoro; //pawn's path

int thesi = -1; //pawn's position in path

boolean thiseas; //true if pawn is thiseas otherwise false

private City city; // pawn’s city

Οι μεθόδοι:

- public Pawn(Path anaktoro); //Constructor

  Creates a new Pawn.

- public Pawn(City city); //Constructor

  Creates a new Pawn.

- public void setThesi(int thesi); //Transformer

  sets the pawn's thesi

- public void setThiseas(boolean thiseas); //Transformer

  sets if pawn is thiseas

- public Path getAnaktoro(); //Accessor

  returns the pawn's path

- public int getThesi(); //Accessor

  returns the pawn's position

- public boolean isThiseas(); //Observer

  Return true if the pawn is thiseas false otherwise

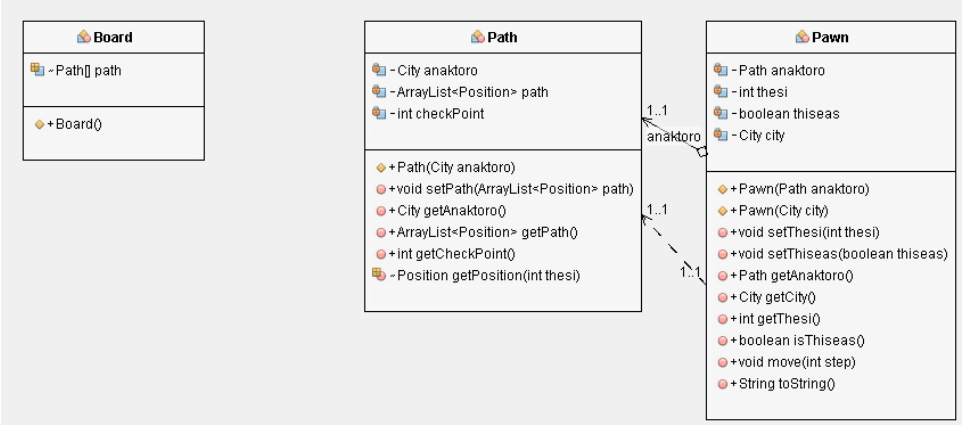
- public void move(int step); //Transformer

  calculate a movement

- public String toString();

  Returns the string representation of a pawn

Uml board



Package position

Class Position

Η κλάση Position υλοποιεί την ύπαρξη μιας θέσης.

Τα attributes:

private int points; //position's points

private int number; //position's number

private String imagePath; //position's image path

Οι μεθόδοι της κλάσης Position:

- public Position(); //Constructor

  Creates a new position.

- public void setPoints (int points); //transformer

  set point in a specific position

- public void setNumber(int number); //transformer

  set position's number

- public int getPoints(); //accessor

  Returns position's points

- public int getNumber(); //accessor

  Returns position's number

Class FindingPosoition

Η κλάση FindingPosoition υλοποιεί την ύπαρξη μιας θέσης με εύρημα.

- public FindingPosition(); //Constructor

  Creates a new position with finding.

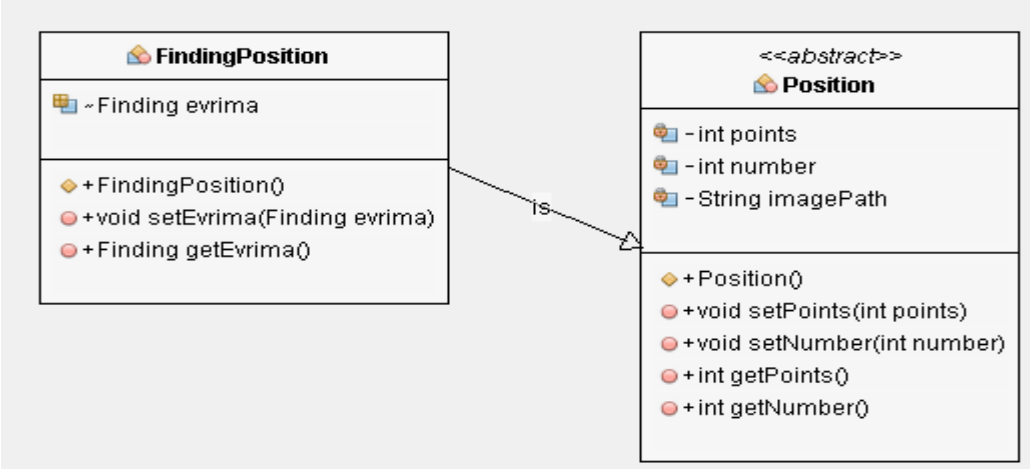
- public void setEvrima(Finding evrima); //transformer

  set position's finding

- public Finding getEvrima(); //accessor

  Returns position's finding

Uml position



Package player

Σε αυτό το πακέτο υπάρχει η κλάση player που υλοποιεί ενα παίκτη.

Class player

Τα attributes:

private String name; // player's name

private int ID; //player's ID

private Sullogi cards; //player's collection of cards in his/her hand

private Sullogi[][] playedCards; //played cards, save played card in Knossos in Sullogi[1][], save played card in Malia in Sullogi[2][],

save played card in Faistosin Sullogi[3][], save played card in Zakrosin Sullogi[4][].

private boolean hasplayed; //says if player has played

private int totalSnakeGoddess; //number of snake goddess which has collected

private ArrayList<Fresco> FrescoClct; //save fresco findings

private ArrayList<RareFinding> RareFindingClct; //save rare findings

private Pawn[] pawn = new Pawn[4]; //player's 4 pawns

Color color; // player's color

Οι μεθόδοι της κλάσης player :

- public player(String name, int ID); //Constructor

  Creates a new player with "name" String and "ID" int.

- public void init_player(); //transformer

  It initializes a player for a new deal(moirasma)

- public int getID(); //accessor

  Returns the ID of a player.

- public void setID(int id); //transformer

  It sets the ID of a player

- public void setID(int id); //transformer

  It sets the ID of a player

- public String getName(); //accessor

  Returns the name of the player

- public void setName(String newName); //transformer

  sets the name of the player to new name

- public Card getCardtoplay(); //accessor

Returns the card who wants a player to play

- public void setCards(Card c); //transformer

  adds a Card to player's cards

- public void Played(); //transformer

  sets the variable hasplayed to true

- public boolean getHasFinished(); //Observer

  Returns if a player has finished the game

- public void addSnakegoddess(); // transformer

  increase by one variable totalSnakeGoddess

- public int getSnakegoddess(); //Observer

  Returns number of snake goddess which player have collected

- public void sethasPlayed(boolean hasplayed); //transformer

  set variable hasplayed with new condition

- public boolean gethasPlayed(); //Observer

  Returns boolean value if player has play

- public void addFrescoClct(Fresco fresco); //transformer

  add a fresco finding in arraylist FrescoClct

- public void addRareFindingClct(RareFinding rarefindinng);
  //transformer

  add a rare finding in arraylist RareFindingClct

- public ArrayList<RareFinding> getRareFindingClct(); //Observer

  Returns a list with player's rare findings

- public Pawn[] getPawns(); //Observer

  Returns player's pawns

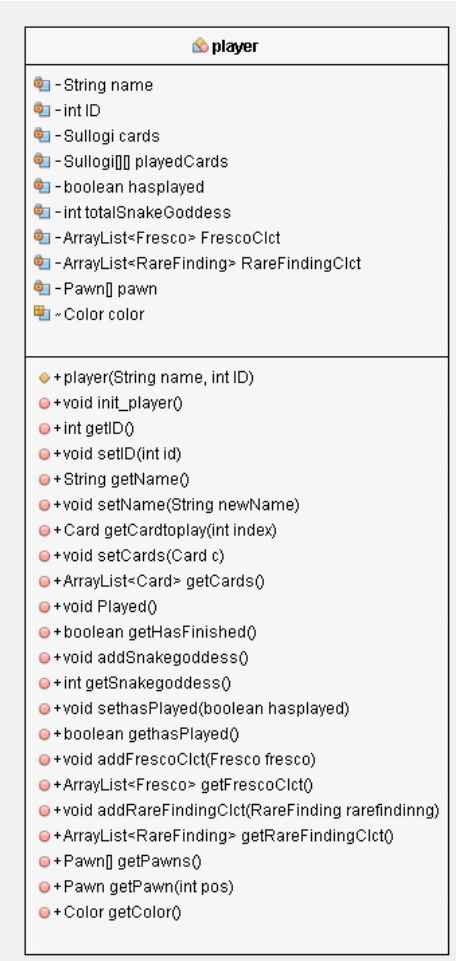
- public Pawn getPawn(int pos); //Observer

  Returns player's specific pawn

- public Color getColor(); //Observer

    Returns player's color

Uml player



Package Controller

Class Controller

Τα attributes:

private player P1, P2; //P1 player and P2 player

private Sullogi allcards = new Sullogi(); //pack of cards

private boolean empty_table, notstarted; //variable empty_table says if table is empty and variable notstarted says if we have start

Sullogi allcards; //pack of cards

Οι μεθόδοι της κλάσης Comtroller:

- public Controller(); //constructor

Constructs a new Controller and sets the game to start

- public void PlayCard(Card card); //transformer

  player play a card

- public boolean CardHasPlayed(); //Observer

  check a card if had played

- public int getTurn(); //accessor

  Returns which player has the turn

- public int setTurn(); //transformer

  set who will play

- public void setStarted(); //transformer

  sets the variable notstarted to false, so game start

- public boolean getStarted(); //Observer

  Return true if the game has not started false otherwise

- public void initPlayerCards(); //transformer

  initializes players cards in the beginning

- public void initTable(); //transformer

  initializes some things(allcards,turn,round) to start game

- public boolean game_over(); //Observer

  Return true if game has finished, false otherwise

- public void initialize(); //transformer

  initializes some things to start game

- private void updatePawns(); // transformer

  update pawns situation

- public void setListeners(); // transformer

  set listeners

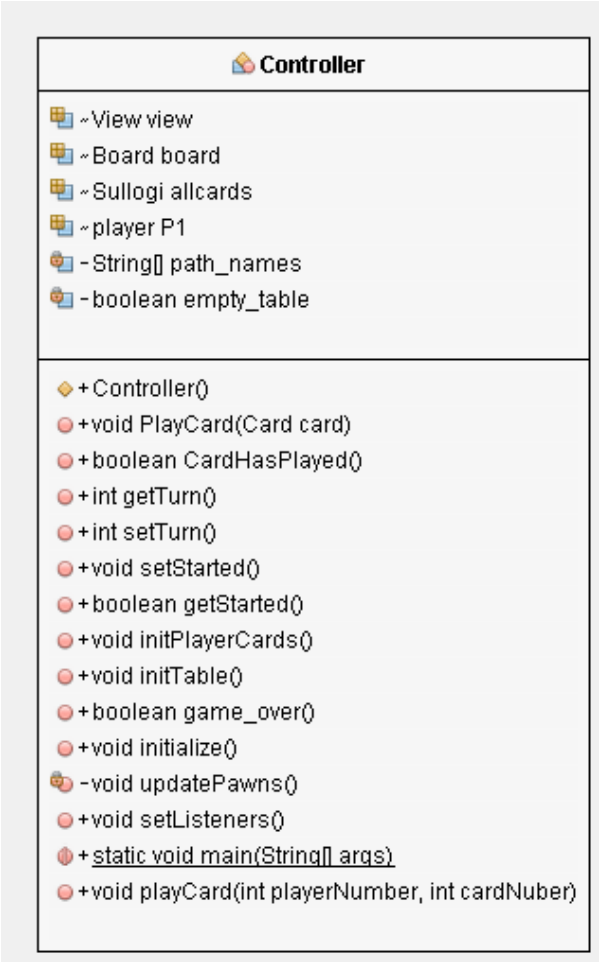
- public static void main(String[] args);

  main for my program

- public void actionPerformed(ActionEvent e); // transformer

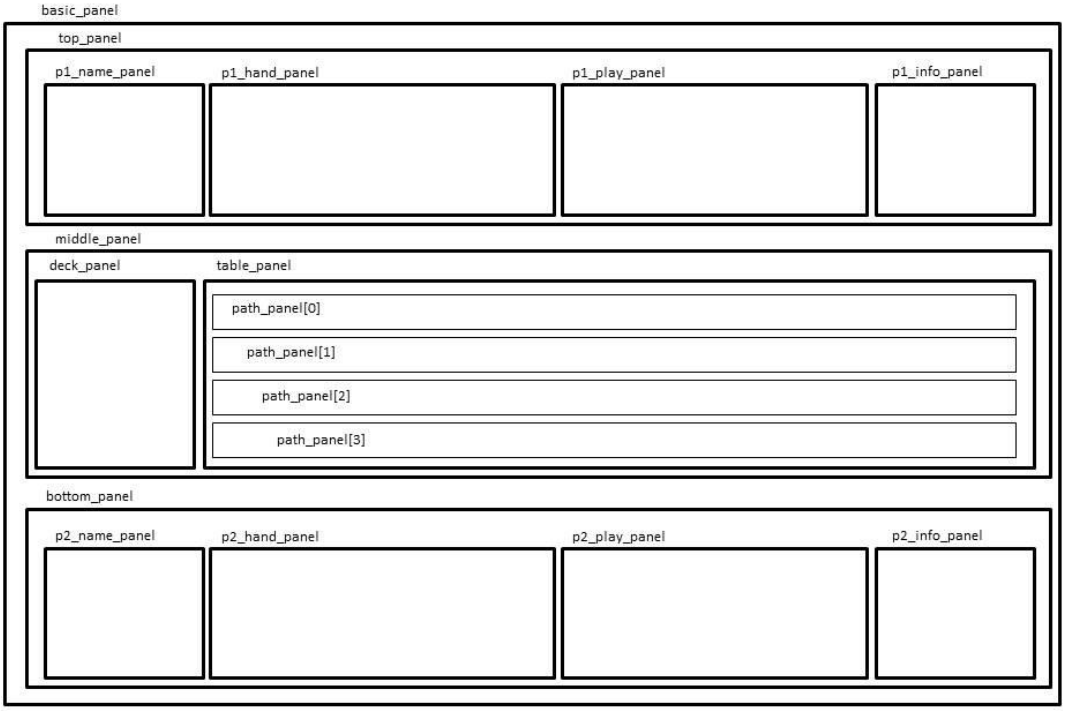
  performe actions

Uml controller



Package View

Αυτό το πακέτο θα αποτελείται από μία κλάση που θα δημιουργεί ένα ένα frame και μέσα σε αυτό ένα panel.Μέσα σε αυτό το panel θα υπάρχουν 2 panels για κάθε παίχτη ,όπου στο πρώτο θα περιλαμβάνονται οι κάρτες του και στο δεύτερο οι παιγμένες κάρτες και τα επιτεύγματα του(δηλαδή πόντοι και ευρήματα). Επίσης θα υπάρχει ένα κεντρικό panel που θα είναι το ταμπλό του παιχνιδιού. Όταν πατιέται μια κάρτα αν είναι η σειρά του παίκτη θα μεταφέρεται στο χώρο των παιγμένων καρτών και θα πραγματοποιούνται οι απαραίτητες αλλαγές στο κεντρικό panel(π.χ. αλλαγή σε θέση πιονιού). Ακόμα θα υπάρχουν κουμπιά για το Quit και New Game και 2 χώροι μηνυμάτων κάτω από τις κάρτες του κάθε παίκτη που θα εμφανίζουν το όνομα και τα πιόνια του κάθε παίκτη.



Στην εικόνα φαίνεται η σχεδίαση του γραφικού περιβάλλοντος .

Έχουμε το basic_panel το οποιο περιέχει τα top_panel, middle_panel, bottom_panel.

Στο top_panel φαίνονται όλες οι πληροφορίες για τον player1, στο οποίο υπάρχουν τα εξής panels: p1_name_panel, p1_hand_panel, p1_play_panel, p1_info_panel.

Στο midle_panel είναι το ταμλό, στο οποίο έχουμε τα deck_panel και table_panel. Στο deck_panel είναι η τράπουλα. Στο table_panel είναι τα τέσσερα μονοπάτια. Τα μονοπάτια έχουν υλοποιηθεί ως κουμπιά για να μπορεί να μετακινούνται τα πιόνια.

Στο bottom_panel φαίνονται όλες οι πληροφορίες για τον player2, στο οποίο υπάρχουν τα εξής panels: p2_name_panel, p2_hand_panel, p2_play_panel, p2_info_panel.

Στο package View έχουμε την κλάση view η οποία έχει τις εξής μεθόδους:

```
public ArrayList<JButton> getPlayerButtons(int player_number);
```

// accessor

Returns player's buttons.

public void clearPawns(); // Transformer

remove all pawns

public void updatePawn(player thePlayer); // Transformer

update pawn

public View(); //constructor

Creates a View

public void initComponents(); // Transformer

initialize all panels

public void showPlayerCards(int playernumber, ArrayList<Card> cards); // Transformer

show player's cards

ΤΕΛΟΣ