



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

HY252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2018-2019

# TICKET TO RIDE

## Φάση Β

*Στοιχεία*

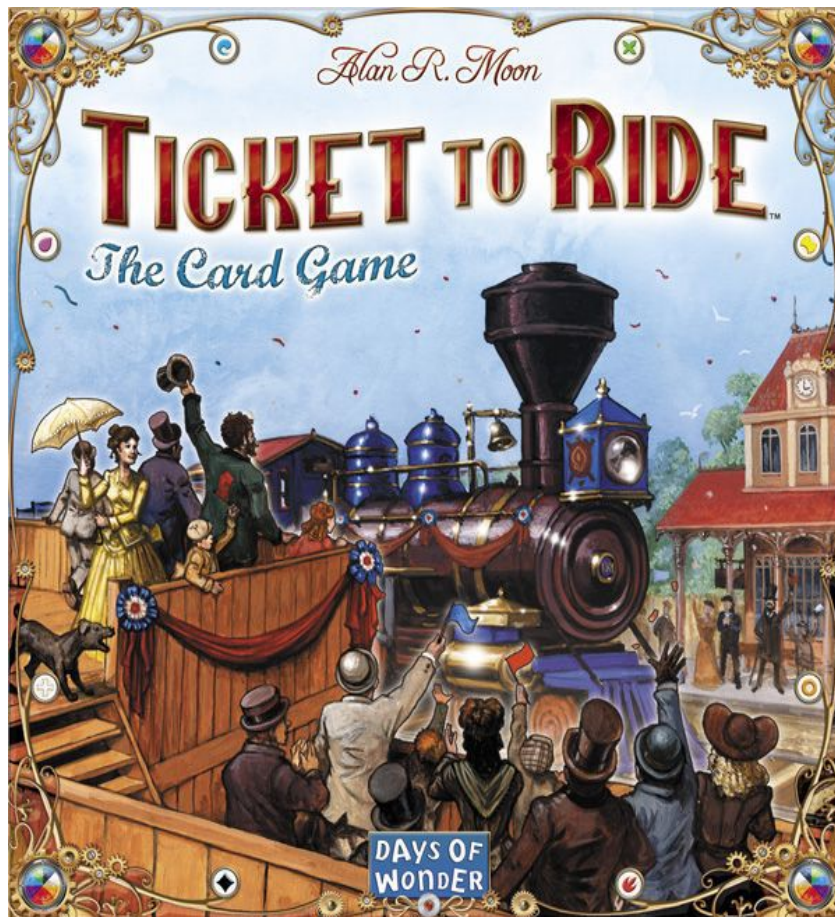
Όνομα: Τζανόπουλος Αλέξανδρος

ΑΜ: math5370

Ημερομηνία: 19 / 01/ 19

## Περιεχόμενα

1. Εισαγωγή.....	1
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model.....	1
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller.....	1
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	2
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....	2
6. Λειτουργικότητα (B Φάση).....	2
7. Συμπεράσματα.....	2



## 1. Εισαγωγή

Η εργασία ακολουθεί το μοντέλο MVC. Έχει χωριστεί, δηλαδή, σε τρία κομμάτια, το Model, το View και το Controller. Ο Controller λειτουργεί σαν συνδετικός κρίκος μεταξύ του Model και του View, χρησιμοποιώντας και εννώνοντας τα στοιχεία και των δύο ώστε να δημιουργηθεί το παιχνίδι.

Στην συνέχεια περιγράφονται πιο αναλυτικά οι λειτουργίες του κάθε μέρους.

## 2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακέτο Model θα περιλαμβάνει κλάσεις για τις Κάρτες, οι οποίες περιλαμβάνονται σε ένα sub-package για λόγους καλύτερης οργάνωσης, για τον Παίκτη, καθώς και για τις διάφορες περιοχές του παιχνιδιού (GameBoard, Rail yards, Tracks). Οι κάρτες ορίζονται σε μια κλάση Card και ορίζονται και οι υποκλάσεις της TrainCard, DestinationCard και BigCityCard. Ο διαχωρισμός αυτός γίνεται επειδή κάθε είδος κάρτας έχει διαφορετικά πεδία.

- Player Interface:

Στον κάθε παίκτη θα δίνουμε ένα Rail yard, Track, ένα χέρι στο οποίο θα κρατάει τις κάρτες που μαζεύει και σε κάθε παίκτη θα κρατάμε ένα σκορ για τους πόντους νίκης που έχει κάθε στιγμή του παιχνιδιού.

- 

Τα πεδία:

- i. private String name;

player's name



- ii. `private String ID;`  
a unique ID for the player
- iii. `private int player_points;`  
Winning points
- iv. `private ArrayList<Card> Hand;`  
The cards in the player's hand
- v. `private Track track;`  
the player's track
- vi. `private RailYard rail;`  
the player's railyard

Οι μέθοδοι:

- i. `public void setName(String name); //setter`
- ii. `public String getName(); //getter`
- iii. `public void setID(int id); //setter`
- iv. `public int getID(); //getter`
- v. `public void addToHand(Card c); //mutator`
- vi. `public ArrayList<Card> getHand() //getter`
- vii. `public RailYard getRail(); //getter`
- viii. `public Track getTrack(); //getter`
- ix. `public Player(  
    String name, int ID, Track track, Rail rail  
); //constructor`



- Card Interface:

Η υπερκλάση Card περιέχει ένα πεδίο για το path της εικόνας της κάρτας.

Τα πεδία:

- i. private String imagePath;  
a path for the card's image

Οι μέθοδοι:

- i. public void setImage(String imgPath);//setter
- ii. public String getImage();//getter
- iii. public Card(String path);//constructor

- TrainCard Interface:

Η TrainCard είναι υποκλάση της Card για τις κάρτες τρένα. Κάθε κάρτα τρένο έχει ένα χρώμα και χρίζεται Locomotive αν το χρώμα το ορίζεται σαν “Locomotive”.

Τα πεδία:

- i. private String color;  
card's color(“Locomotive” for locomotive)



Οι μέθοδοι:

- i. `public boolean isLocomotive(); //observer`
- ii. `public void setColor(String color); //setter`
- iii. `public String getColor();//getter`
- iv. `public TrainCard(String path, String color);//Constructor`

- DestinationCard Interface:

Η DestinationCard είναι υποκλάση της Card για κάρτες προορισμού. Κάθε κάρτα προορισμού έχει μια λίστα χρωμάτων, πόντους, άφιξη και προορισμό.

Τα πεδία:

- i. `private ArrayList<String> colors;`  
The list of colors in the card
- ii. `private int points;`  
points worth
- iii. `private String from;`  
starting post
- iv. `private String to;`  
destination



Οι μέθοδοι:

- i. `public void setColors(ArrayList<String> colors); //setter`
- ii. `public ArrayList<String> getColors; //getter`
- iii. `public void addColor(String color); //mutator`
- iv. `public int pointValue(); //accessor`
- v. `public void setStart(String start); //setter`
- vi. `public String getStart(); //getter`
- vii. `public String getDest(); //getter`
- viii. `public DestinationCard(  
    String path, ArrayList<String> colors, int points, String  
    from, String to  
); //Constructor`

- BigCityCard Interface:

Η BigCityCard είναι υποκλάση της Card για κλάσεις πόλεων.  
Κάθε BigCity έχει όνομα, πόντους και φορές επίσκεψης.

Τα πεδία:

- i. `private String cityName;`
- ii. `private int points;`  
`points worth`
- iii. `private int timesVisited;`



Οι μέθοδοι:

- i. `public void setName(String name); //setter`
- ii. `public String getName(); //getter`
- iii. `public int pointValue(); //accessor`
- iv. `public int Visisted(); //accessor`
- v. `public BigCityCard(  
    String path, String name, int points  
); //Constructor`

• RailYard Interface:

Η RailYard κλάση περιγράφει το RailYard του παίχτη. Περιέχει 8 στοίβες με κάρτες, μια για κάθε χρώμα τρένου.

Τα πεδία:

- i. `ArrayList<Card> redPile = new ArrayList<Card>();`
- ii. `ArrayList<Card> blackPile = new ArrayList<Card>();`
- iii. `ArrayList<Card> bluePile = new ArrayList<Card>();`
- iv. `ArrayList<Card> greenPile = new ArrayList<Card>();`
- v. `ArrayList<Card> purplePile = new ArrayList<Card>();`
- vi. `ArrayList<Card> whitePile = new ArrayList<Card>();`
- vii. `ArrayList<Card> yellowPile = new ArrayList<Card>();`
- viii. `ArrayList<Card> orangePile = new ArrayList<Card>();`





Οι μέθοδοι:

- i. `public ArrayList<Card> getRedCards(); //getter`
- ii. `public ArrayList<Card> getBlackCards(); //getter`
- iii. `public ArrayList<Card> getBlueCards(); //getter`
- iv. `public ArrayList<Card> getGreenCards(); //getter`
- v. `public ArrayList<Card> getPurpleCards(); //getter`
- vi. `public ArrayList<Card> getWhiteCards(); //getter`
- vii. `public ArrayList<Card> getYellowCards(); //getter`
- viii. `public ArrayList<Card> getOrangeCards(); //getter`
- ix. `public RailYard(  
    ArrayList<ArrayList<Card>>piles  
); //Constructor`

- Track Interface:

Η κλάση Track είναι, σχεδιαστικά, υποκλάση της RailYard διότι περιέχει τα ίδια πράγματα συν μιάς στοίβας για τρένα τύπου Locomotive.

Τα πεδία:

- i. `private ArrayList<Card> locomotivePile;`

Οι μέθοδοι:

- i. `public ArrayList<Card> getLocomotivePile();//getter`
- ii. `public void setLocomotivePile();//observer`



```

iii. public Track(
        ArrayList<ArrayList<Card>> colorPiles,
        ArrayList<Card> locPile
    );//Constructor
    
```

- Deck Interface:

Η κλάση Deck αναπαριστά την τράπουλα του παιχνιδιού. Η Deck, ουσιαστικά, είναι μια στοίβα αντικειμένων Card. Η Deck θα χρησιμοποιηθεί ώστε να φτιαχτούν τράπουλες για τις κάρτες τρένα και τις κάρτες προορισμού.

Τα πεδία:

- i. private ArrayList<Card> cards;  
the cards contained in the deck

Οι μέθοδοι:

- i. public boolean isEmpty();//observer  
return true if deck is empty; false otherwise
- ii. public int cardsLeft();//observer
- iii. public ArrayList<Card> getCards();//getter
- iv. public void addCardToDeck(Card c, int position);//mutator
- v. public Card getTopCard();//getter
- vi. public void removeTopCard();//mutator
- vii. public Deck(ArrayList<Card> cards);//Constructor



- GameBoard Interface:

Η κλάση GameBoard περιγράφει το τραπέζι ανάμεσα τους 2 παίκτες . Περιέχει 2 τράπουλες, μια για τις κάρτες τρένα και μια για τις κάρτες προορισμού, μια λίστα με κάρτες τρένα και μια λίστα με τις κάρτες μεγάλων πόλεων. Στην κλάση αυτή υπάρχουν επίσης μέθοδοι για την δημιουργία των τράπουλών.

Τα πεδία:

- i. private Deck trainCards;
- ii. private Deck destCards;
- iii. private ArrayList<Card> centerCards;
- iv. private ArrayList<Card> bigCityCards;

Οι μέθοδοι:

- i. public Deck getTrainCards();//getter
- ii. public Deck getDestCards();//getter
- iii. public ArrayList<Card> getCenterCards();//getter
- iv. public ArrayList<Card> getBigCityCards();//getter
- v. public GameBoard(  
         Deck trains, Deck dest, ArrayList<Card> center,  
         ArrayList<Card> bigCity  
     );//Constructor



### 3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Το πακέτο Controller θα περιέχει την κλάση Controller η οποία θα είναι υπεύθυνη για όλες τις ενέργειες του παιχνιδιού, χρησιμοποιώντας τις κλάσεις του πακέτου Model και View. Η κλάση αυτή θα υλοποιεί ενέργειες όπως το μοίρασμα καρτών, την απόδοση πόντων νίκης, τον έλεγχο σειράς των παιχτών, τον έλεγχο τερματισμού του παιχνιδιού κλπ.

- Controller Interface:

Τα πεδία:

- i. private Player P1, P2;
- ii. private GameBoard gb;
- iii. private boolean choiceMade;
- iv. private int playerPlaying;  
    playing player's ID

Οι μέθοδοι:

- i. public void dealCards(); //mutator



- ii. `public void playerTurn(Player P); //observer`
- iii. `public void moveToTrack(); //mutator`
- iv. `public void drawTrains(); //mutator`
- v. `public void drawDest();//mutator`
- vi. `public void transferToRail();//mutator`
- vii. `public void buyDest();//mutator`
- viii. `public boolean canMovetoRail(  
                    ArrayList<Card> from Hand  
                    );//observer`
- ix. `public boolean gameFinished();//observer`
- x. `public void announceWinner();//observer`
- xi. `public void tranRob();//mutator`
- xii. `public boolean canBuy(DestinationCard dc);//observer`
- xiii. `public void updateScore();//observer`
- xiv. `public void giveBigCityCard();//mutator`
- xv. `public Player getWinner();//getter`
- xvi. `public Player Tiebreak();//getter`
- xvii. `public GameBoard getGameBoard();//getter`

## 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View θα περιέχει κλάσεις για την δημιουργία του γραφικού περιβάλλοντος του παιχνιδιού. Οι κλάσεις αυτές θα δημιουργούν την



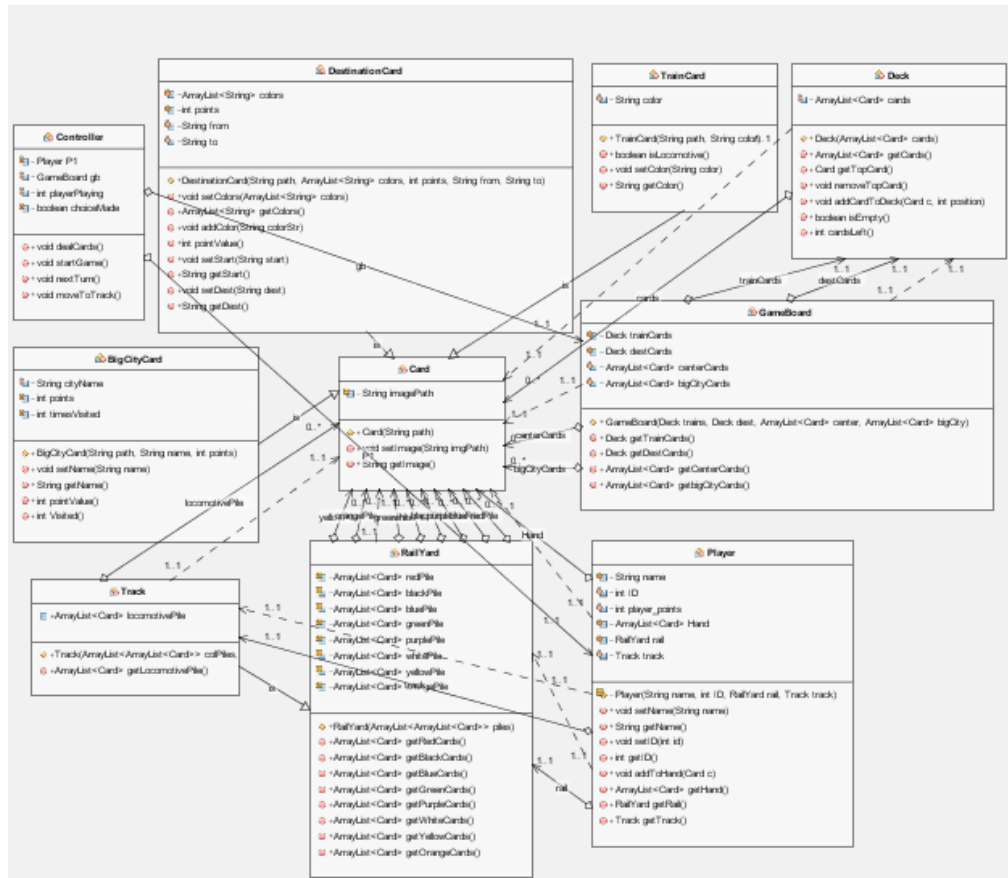
γραφική μορφή των GameBoard, Track, RailYard και hand του Player, τα κουμπιά τα οποία θα χρειάζονται καθώς και θα υλοποιούν τις ενέργειες οι οποίες θα γίνουν όταν πατηθεί το κάθε κουμπί.

Το γραφικό περιβάλλον θα περιλαμβάνει ένα Frame το οποίο θα περιέχει Panels για το κάθε RailYard, Track και για το GameBoard. Τα εξής Panels θα περιέχουν κουμπιά για μεταφορά καρτών.

Το

## 5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

### Το ολικό διάγραμμα UML:



## 6. Λειτουργικότητα (B Φάση)

Δεν κατάφερα να υλοποιήσω τα 2 διπλανά κουμπιά, και ενδεχομένως να υπάρχει ένα πρόβλημα στο train robbing. Επιπλέον, παρέλειψα τα εικονίδια στα Tracks, λόγω χρόνου

## 7. Συμπεράσματα

Το πιο δύσκολο κομμάτι ήταν να βρώ πώς και απο πού θα ξεκινήσω, αλλά απο την στιγμή που άρχισα να καταλαβαίνω καλύτερα την λογική του συστήματος MVC, όλα έπεσαν στην θέση τους. Δεν χρησιμοποίησα αρκετά πράγματα απο την πρώτη φάση, και πιστεύω οτι θα έπρεπε να είχα ξεκινήσει απο τότε την υλοποίηση, παρά να φτιάξω ένα abstract μοντέλο το οποίο δεν ήξερα πως θα χρησιμοποιήσω. Μετά απο την υλοποίηση όμως, πήρα αρκετή εμπειρία και ξέρω πλέον πως να χρησιμοποιώ αυτο το μοντέλο, και θα το χρησιμοποιήσω σε μελλοντικά project

