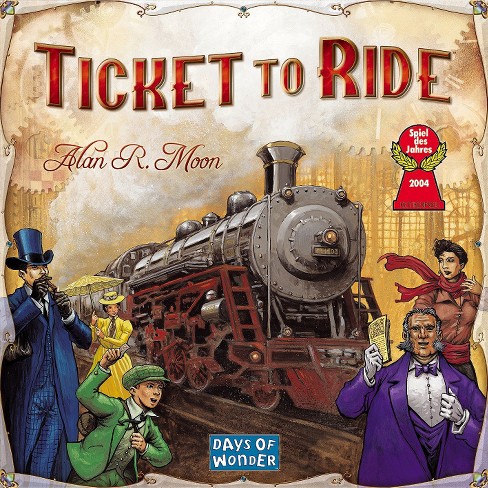
PROJECT

ΕΙΣΑΓΩΓΗ

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller).Έτσι , σκοπός μας είναι ο Controller να συνδεει 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.

Package Model

Σε αυτό το πακέτο θα περιέχονται διεπαφή Card,οι κλάσεις CardColor,CardFace ,TrainCard,BigCitiesCard,CardType,DestinationCard,PointCard κλάσεις που κληρονομούν την Card , η κλάση Player , η κλάση Turn.

**Card Interface and Other Classes for Cards**

Αρχικά φτιάχνοντας τη διεπαφή Card μας δίνεται η δυνατότητα να προσπελάσουμε τα δεδομένα χωρίς να πρέπει να ορίσουμε αν μία κάρτα είναι train,destination.

Το interface αυτό μας παρέχει τις εξής μεθόδους:

public void setface(CardFace f);

1. public void settypecard(CardType f);

Sets the type of the card

2. public CardFace getface(CardFace f);

set face of card

3. public boolean setActivation(boolean active);

set activation of a card

4. public boolean getActivation() ;

Get if it is active

5. public int PlayerBelongs() ;

returns in who player the card belongs

6. public int getValue();

returns the card's value

7. public void setValue(int value);

sets the card's value

public int getPoints();

Returns the points of a card

8. public void setPoints(int points);

sets the points of a card

9. public String toString();

Returns the string representation of a card

10. public void SetTempValue(int j);

sets the card's temp value

11. public int GetTempValue();

returns the card's temp value

**CARD COLOR**

Είναι ένα απλο enum το οποιο μας αναφερει το τι χρωμα εχει η καρτα

**CARD FACE**

Μας λεει για το αν μια καρτα είναι γυρισμενη ή όχι

(ένα enum και αυτό)

**CARD TYPE**

Μας λεει για το τι είναι η καρτα δλδ train card,…

(enum)

**DESTINATION CARD**

1.public void setList(TrainCard a)

Προσθετει μια καρτα για την λιστα με το τι χρειαζεται

2. public int getPoints() επιστρεφει τους ποντους

3. public boolean checkList(ArrayList<TrainCard> b)

Ελενχει αν οι 2 λιστες είναι ιδιες με βαση τα στοιχεια

**BIG CITIES CARD**

1.Μετραει ποσες φορες καλειται από τον παικτη 1 και τον 2 ξεχωριστα

public void called1() {

visit1++;

}

public void called2() {

visit2++;

}

2.επιστρεφει ποσες φορες καλεστηκε από τον κάθε παικτη

public int s1() {

return visit1;

}

public int s1() {

return visit1;

}

**Points Card**

1.public boolean setActivation(boolean active) κανει την καρτα active

2. public boolean getActivation() επιστρεφει αν η καρτα είναι active

3. public int PlayerBelongs() επιστρεφει στον παικτη που ανηκει

4. public int getValue() επιστρεφει την αξια της κάθε καρτας

5. public void setValue(int value) θετει την αξια της κάθε καρτας

6. public int getPoints() επιστρεφει τους ποντους της καρτας

7. public void setPoints(int points) θετει τους ποντους της καρτας

8. public void SetTempValue(int j) επιστρεφει την temp value

9. public int GetTempValue()θετει την temp value

10. public void setface(CardFace f) θετει το enum της καρτας

11. public CardFace getface(CardFace f) παιρνει το enum της καρτας

12. public void settypecard(CardType f) θετει το type της καρτας

13. public CardType gettypecard() επιστρεφει το type της καρτας

**Train Card**

1. public int getValue() επιστρεφει την τιμη της καρτας

2. public CardColor getColor() επιστρεφει το χρωμα της καρτας

3. public void setValue(int value)θετει την τιμη της καρτας

4. public int getPoints() παιρνει τους ποντους κάθε καρτας

5. public void setPoints(int points) θετει τους ποντους κάθε καρτας

6. public void SetTempValue(int j) θετει το temp κάθε καρτας

7. public int GetTempValue()διαβαζει το temp κάθε καρτας

8. public boolean setActivation(boolean active) θετει την καρτα αν είναι active

9. public boolean getActivation() επιστρεφει το active κάθε καρτας

10. public int PlayerBelongs() επιστρεφει σε ποιον παικτη ανοικει

11. public void setface(CardFace f) θετει το CardFace σαν up ή down

12. public CardFace getface(CardFace f) επιστρεφει το cardface

13. public void settypecard(CardType f) θετει το CardType σαν up ή down

14. public CardType gettypecard() επιστρεφει το cardType σαν up ή down

**MODEL.PLAYER**

Σε αυτό το package εχω μονο την κλαση player

**PLAYER**

1. it starts to shuffles the cards

public void shuffleCards()

2. It initializes a player for a new deal(moirasma)

public void init\_player()

3.Returns the name of the player

public String getName()

4.sets the name of the player to newName

public void setName(String newName)

5. gets the player extra points

public void getPoints(int points)

6. sets the collection(ArrayList<TrainCard>) who wants a player to play to cardsforplaying

public void setCards\_to\_play(ArrayList<TrainCard> cardsforplaying)

7.Returns the cards who wants a player to play <br />

public ArrayList<TrainCard> getCards\_to\_play()

8. adds a Card to players cards <br />

public void setCards(Card c)

9. adds a collection to players points(cards which count for points)

public void set\_points(ArrayList<TrainCard> s)

10. Returns the Cards collection of a player <br />

public ArrayList<TrainCard> getCards()

11.sets the variable hasplayed to true

public void Played()

12. Sets the variable has\_finished to true

public void has\_finished()

13. Returns if a player has played at least one time

public boolean Has\_Played()

14.Returns if a player has finished the game(partida)

public boolean Get\_has\_finished()

**TURN**

1. Sets the player's turn.(which player has the turn to play)

public void setID(ArrayList <Player> players)

2. returns the player's ID whose turn is to play

public int getID()

3.Checks if a player has finished

public boolean checkIfPlayerFinished(Player p)

4. Sets the number of players.

public void NumberOfPlayers()

{

}

5. returns the number of players

public int GetNumberOfPlayers()

6. Sets the most recent player, who has dropped cards in the table.

public void Set\_last\_player(int k)

7. returns the most recent player, who has dropped cards in the table

public int Get\_last\_player()

**MODEL.RAILYARD**

**ON THE TRACK**

1.collect cards from railyard

public void collectFromRailYard(Railyard R,ArrayList<TrainCard> cards)

2. take a card and put it on railyard

public Railyard buyDestinationCard(Railyard R,Card C)

3. makes the train robbing of players

public Railyard trainRobbing(Railyard R,ArrayList<TrainCard> cards)

4.play a card from train card to railyard

public void playCards(ArrayList<TrainCard>A,Railyard R)

5. Put the card on the track

public void OnTheΤrack(Railyard R,ArrayList<TrainCard>A)

6.ckeck for train robbing

public boolean trainRodding(ArrayList<TrainCard>A,Railyard R)

**CONTROLLER**

Αυτή η κλάση είναι ουσιαστικά το μυαλό του παιχνιδιού. Είναι υπεύθυνη για

τη δημιουργία ενός νέου παιχνιδιού, μιας νέας παρτίδας , τη δημιουργία

στιγμιοτύπων παικτών, ομάδων σειράς και γύρων και φυσικά τη σύνδεση

μεταξύ των γραφικών και του Model.Αυτό που κάνει η κλάση αυτή είναι να

παίρνει τις επιλογές του χρήστη μέσω των γραφικών και να πραγματοποιεί

οποιαδήποτε ενέργεια χρειάζεται έτσι ώστε το παιχνίδι να παίζεται σωστά.

Φυσικά είναι υπεύθυνη αυτή η κλάση για να υπολογίζει το σκορ και να

ενημερώνει πότε τελειώνει το παιχνίδι.

1.Return true if the collection that player wanted to play

public boolean GetCollectionHasPlayed()

2.take an integer arraylist from view Class

public void PlayCollection(ArrayList<Integer> cardsPosition)

3.Returns which player has the turn

public int seeTurn()

4.if a player press the button fold ,it increases

\* the variable fold (fold++) or sets the variable fold to 0 if a round finished

public void set\_Fold()

5.Return true if the table is empty false otherwise

public boolean tableIsEmpty()

6. sets the variable not\_started to false

public void set\_started()

7. Return true if the game has not started false otherwise

public boolean not\_started()

8.increases the variable isready by 1(isready++)

public void isready()

9. sort the cards for each player

public void sort\_cards()

9. sets the score of a game after one deal(partida) has finished

public void setScore()

10.Returns the score of the game <br />

public int GetScore()

11. initializes players cards in the beginning

public void init\_player\_cards()

12. initializes some things(allcards,turn,round) for a new deal(partida)

public void init\_table()

13. Return true if we have a new round false otherwise

public boolean Get\_new\_round()

14.make changes after a round started

public void makeChanges(ArrayList<Integer> cardsPosition)

15.Return true if a deal(partida) has finished, false otherwise

public boolean partida\_has\_finished()

16.Return true if a game(one team reaches 1000 points) has finished, false otherwise

public boolean game\_has\_finished()

17.give the turn to player1

public void set\_Turn()

**Package View**

Αυτό το πακέτο θα αποτελείται από μία κλάση που θα δημιουργεί ένα ένα

frame και μέσα σε αυτό 8 panel.Μέσα σε αυτό το panel θα υπάρχουν 4

panels για κάθε παίχτη ,όπου στο πρώτο θα περιλαμβάνονται οι κάρτες του

και στο δεύτερο η συλλογή που θέλει να παίξει στο άλλο οι καρτες που εχει μαζεψει και οι καρτες για το destination επίσης θα υπάρχει ένα

κεντρικό panel που θα είναι το ταμπλό του παιχνιδιού και κουμπιά τις καρτες. Ακόμα θα υπάρχουν κουμπιά

που θα αντιστοιχούν στις κάρτες του παιχνιδιού (πχ το κουμπί 0 θα

αντιστοιχεί στην κάρτα που βρίσκεται στη θέση 0 στη συλλογή με όλες τις

κάρτες).