Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

Περιεχόμενα

- Εισαγωγή

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC(Model View Controller). Έτσι ο σκοπός μας είναι ο Controller να είναι ο συνδετικός κρίκος των Model και View. Οπότε στην 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 λίγο στο View.

- Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Σε αυτό το πακέτο θα περιέχονται abstract classes Tile,Character, κλάσεις που κληρονομούν την κλάση Tile: LandslideTile,FindingTile, κλάσεις που κληρωνομούν την κλάση FindingTile: AmphoraTile, MosaicTile, SkeletonTile, StatueTile, κλάσεις που κληρωνομούν την κλάση StatueTile: SphinxTile,CaryatidTile, κλάσεις που κληρωνομούν την κλάση Character: Assistant, Digger, Archaeologist, Professor και τέλος περιέχει τις κλάσεις Turn, Bag, Board, Player, B_Position.

Tile Interface and other classes for Tiles.

- **Abstract Class Tile:**

Αρχικά η διεπαφή Tile χωρίζεται σε υποκλάσεις ώστε να μας δίνεται η δυνατότητα να έχουμε ευκολότερη πρόσβαση σε κάθε είδος Tile.

Το interface αυτό περιέχει 3 Constructors:

Tile(String category);

Αυτός ο constructor θέτει τιμές στο είδος του Tile.

Αναφέρεται αποκλειστικά και μόνο στα StatueTiles και τα LandslideTiles αφού δεν έχουν ούτε χρώμα ούτε κομμάτια(skeleton parts). Η διαδικασία γίνεται ως εξής: όταν έχουμε ένα CaryatidTile ή ένα SphinxTile επιστρέφουν στην κλάση StatueTile το String "CaryatidTile" ή "SphinxTile" αντίστοιχα και η StatueTile με την σειρά της επιστρέφει το String αυτό στην κλάση FindingTile (θα μιλήσουμε αργότερα για αυτήν) και αυτή με την σειρά της στον Constructor Tile(category).

Tile(String category,String colour);

Αυτός ο Constructor θέτει τιμές στο είδος και στο χρώμα του Tile.

Αναφέρεται αποκλειστικα και μονο στα AmphoraTile και MosaicTile αφού είναι τα μόνα είδη tiles με χρώμα. Η διαδικασία είναι παρόμοια με την παραπάνω με την διαφορά ότι τώρα οι κλάσεις AmphoraTile και MosaicTile εκτός από το είδος επιστρέφουν και το χρώμα του tile.

Tile(String category,String part,int temp);

Αυτός ο Constructor θέτει τιμές στο είδος και στα κομμάτια των tiles.

Αναφέρεται αποκλειστικά και μόνο στα SkeletonTiles τα οποία χωρίζονται σε 4 κομμάτια(πάνω και κάτω μέρη μεγάλου σκελετού και πάνω και κάτω μέρη μικρού σκελετού). Η διαδικασία γίνεται ώς εξής:

Αρχικά για να δείξουμε στο πρόγραμμα ποιό είδος κομματιού σκελετού θέλουμε να δώσουμε στην Tile χρησιμοποιούμε έναν integer (το 1 είναι το πάνω μέρος μεγάλου σκελετού το 2 είναι το κάτω μέρος μεγάλου σκελετού το 3 είναι το πάνω μέρος του μικρού σκελετού και το 4 το κάτω μέρος του μικρού σκελετού).

Έπειτα η SkeletonTile με παρόμοιες διαδικασίες με τα παραπάνω επιστρέφει στην Tile(String category,String part,int temp) το είδος του Tile και το κομμάτι του σκελετού που ζητήσαμε.

(Το int temp είναι για να ξεχωρίζουμε τους 2 Constructors (λόγω του ότι έχουν και οι 2 παραμέτρους 2 Strings):

Tile(String category,String part,int temp), Tile(String category,String colour).

<u>Το interface αυτό μας παρέχει τις εξής μεθόδους:</u>

1. public String getcategory(); Accessor(Selector)
   Returns the value of the category of a tile.
2. public String getcolour(); Accessor(Selector)
   Returns the value of the colour of a tile (for Amphora and Mosaic tiles).
3. public String getpart(); Accessor(Selector)
   Returns the value of the part of a tile (for skeleton tiles).

<u>Attributes:</u>

1)private String category; //the category of a tile

2)private String colour; //the colour of a tile

3)private String part; //the part of a skeleton tile

Classes FindingTile, LandslideTile

Αυτές οι κλάσεις κάνουν extend την Tile και μέσω της εντολής super αποκτούν πρόσβαση στην κλαση Tile και αρχκοποιούν τις τιμές category colour και part.

- **Class FindingTile**

Attributes: NONE

Constructors:

FindingTile(String category)

FindingTile(String category,String colour)

FindingTile(String category,String part,int k)

Οι παραπάνω constructors λειτουργούν όπως ακριβώς και στα Tiles που αναφέρεται πιο πάνω (με μία διαφορά στον 1ο constructor ο οποίος δεν παίρνει LandslideTiles αλλά μόνο StatueTiles).

Η χρήση αυτής της κλάσης είναι καθαρά βοηθητική καθώς ξεχωρίζει τα πλακίδια που παίρνουν πόντους με αυτά που δεν παίρνουν(LandslideTiles).

- **Class LandslideTile**

Attributes: None

Constructor:

Landslide()

Calls the super class Tile with parameter the string “Landslide”

Methods: None

Classes AmphoraTile, MosaicTile,StatueTile,SkeletonTile

Αυτές οι κλάσεις κάνουν extend την FindingTile και μέσω της εντολής super αποκτούν πρόσβαση στην κλαση FindingTile.

- **Class AmphoraTile**

Attributes:

1) private static String colour; //the colour of an Amphora tile

Constructor:

AmphoraTile(int c)

Calls the super class FindingTile with parameters the string "AmphoraTile" and the colour of that tile

Methods:

public static String setcolour(int c); Accessor(Selector)

Gets the colour of a tile according to an integer (1 = blue, 2 = brown, 3 = red, 4 = green, 5 = yellow, 6 = purple).

- **Class MosaicTile**

Attributes:

1)private static String colour;// the colour of a Mosaic tile

Constructor:

MosaicTile(int c)

Calls the super class FindingTile with parameters the colour and the category of a tile (in this case "MosaicTile")

Methods:

1. public static String setcolour(int c); Accessor(selector)
   Gets the colour of a tile according to an integer and returns its value(1 = green, 2 = red, 3 = yellow)

- **Class SkeletonTile**

Attributes:

1)private static String part; // the part of a skeleton tile.

Constructor:

SkeletonTile(int p)

Calls the super class FindingTile with parameters the string “SkeletonTile” and the part of the skeleton as a string)

Methods:

1.public static String getpart(int p); Accessor(selector)

Gets the part of a skeleton tile according to an integer and returns its value (1=Big_UP, 2=Big_DOWN,3=Small_UP,4=Small_DOWN).

- **Class StatueTile**

Attributes: None

Constructor:

StatueTile(String category)

Calls the super class FindingTile with parameter the String category

Methods: None

Classes CaryatidTile, SphinxTile

Αυτές οι κλάσεις κάνουν extend την StatueTile και μέσω της εντολής super αποκτούν πρόσβαση στην κλαση StatueTile.

- **Class CaryatidTile**

Attributes: None

Constructor:

CaryatidTile()

Calls the super class StatueTile with parameter the String “CaryatidTile”

Methods: None

- **Class SphinxTile**

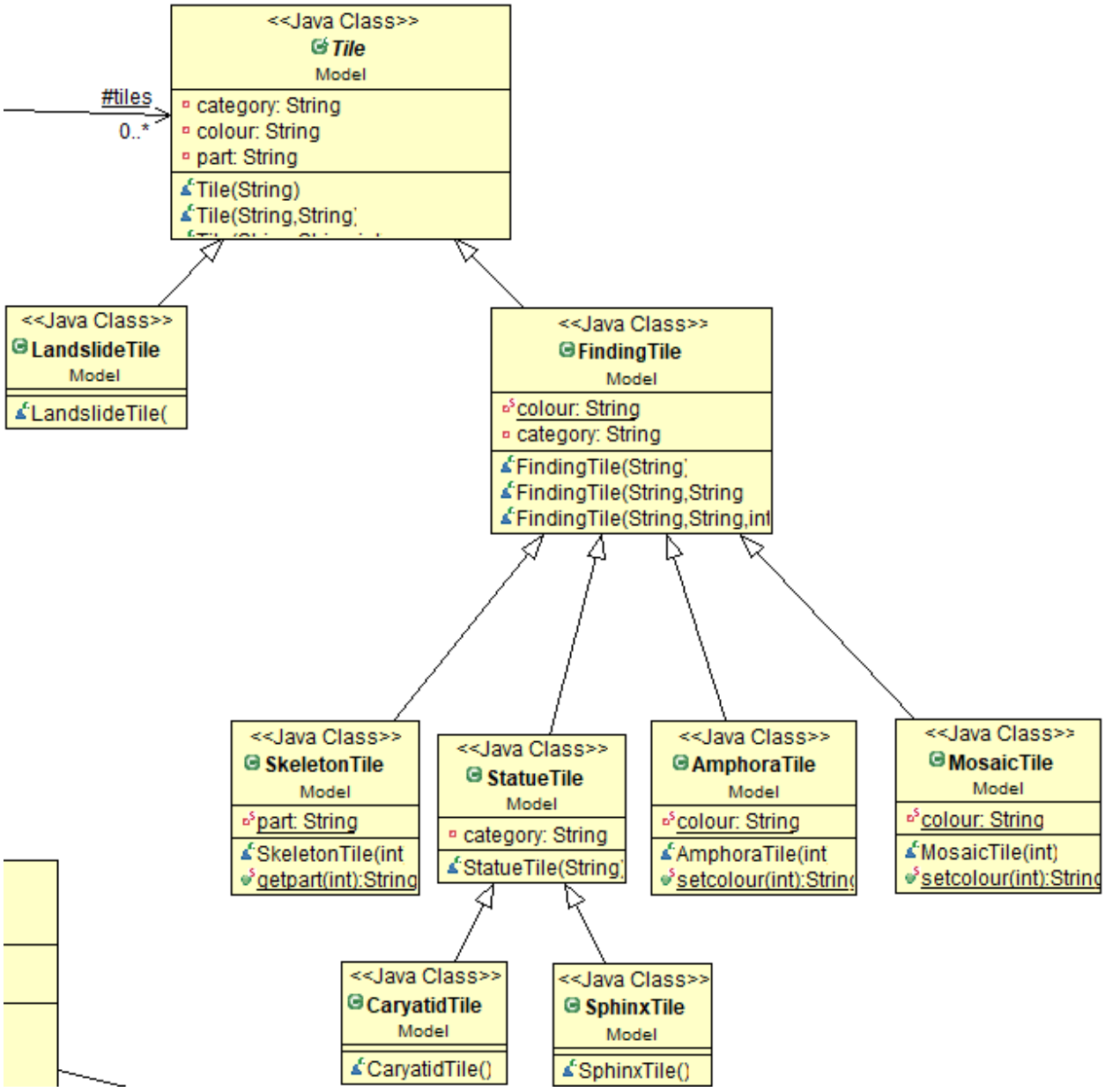
Attributes: None

Constructor:

SphinxTile()

Calls the super class StatueTile with parameter the String “SphinxTile”

Methods: None



- **Class Bag**

  Η κλάση αυτή περιέχει όλα τα tiles του παιχνιδιού (135) σε ένα ArrayList τύπου Tile και με αυτόν τον τρόπο μπορούμε εύκολα να διαχειριζόμαστε τα tiles (διαγραφή ενός tile κτλ).

Attributes:

1)protected static ArrayList<Tile> tiles; // A List with all the Tiles in it.

Constructor:

Bag()

Initializes every single tile (all 135) and puts them on a list (a bag).

This constructor calls every tile method and constructor in order to initialize the tiles.

Methods: None

- **Class Character**
  Είναι μια abstract μέθοδος η οποία χωρίζεται για λόγους ευκολίας στις υποκλάσεις:
  1)Assistant
  2)Digger
  3)Archaeologist
  4)Professor

Attributes:

1)private String colour; // the colour of the Character's card

2)private Boolean Used = false; // a Boolean variable which is false when the card hasn't been played yet.

3)private String Player; // the name of the player who has this card

4)private String category // the category of the card (the subclasses of the class Character are the categories)

Constuctor:

Character(String category,String colour)

This constructor sets the value of the category and the colour of a Character's card (according to the class Character subclasses).

Methods:

1)public Boolean getUsage(); //Observer

Returns the value of the Boolean variable Used in order to check if this card is used or not.

2)public String getPlayer();//Accessor

Returns the name of the player who has the Character's card.

3)public void USED()//Transformer

Sets the value of the Boolean variable Used to true. This method is usefull only when the card it's reffering to, has already been used.

4)public String getcategory()//Accessor

Returns the value of the string category, in order to check the category of a Character's card.

5)public String getcolour();//Accessor

Returns the value of the string colour, in order to check the colour of a Character's card.

Classes Assistant,Archaeologist,Digger,Professor

Αυτές οι κλάσεις κάνουν extend την κλάση Character και μέσω της εντολής super

Αποκτούν πρόσβαση στην κλάση Character και αρχικοποιούν τις τιμές colour και category.

- **Class Assistant**

Attributes:

1)private static String colour; // the colour of a Character's card.

Constructor:

Assistant(int c)

Calls the super class Character with parameters the string "Assistant" and the string colour of an Assistant card.

The integer c is for the colour of the card.

Methods:

1)public static String setcolour(int c);//Accessor(selector)

Sets the colour of an Assistant card and returns it's value.

2)public void Power()//Transformer

Sets the capabilities of the Assistant card.

- **Class Archaeologist**

Attributes:

1)private static String colour; // the colour of a Character's card.

Constructor:

Archaeologist(int c)

Calls the super class Character with parameters the string "Archaeologist" and the string colour of an Archaeologist card.

The integer c is for the colour of the card.

Methods:

1)public static String setcolour(int c);//Accessor(selector)

Sets the colour of an Archaeologist card and returns it's value.

2)public void Power()//Transformer

Sets the capabilities of the Archaeologist card.

- **Class Digger**

Attributes:

1)private static String colour; // the colour of a Character's card.

Constructor:

Digger(int c)

Calls the super class Character with parameters the string "Digger" and the string colour of a Digger card.

The integer c is for the colour of the card.

Methods:

1)public static String setcolour(int c);//Accessor(selector)

Sets the colour of a Digger card and returns it's value.

2)public void Power()//Transformer

Sets the capabilities of the Digger card.

- **Class Professor**

Attributes:

1)private static String colour; // the colour of a Character's card.

Constructor:

Professor(int c)

Calls the super class Character with parameters the string “Professor” and the string colour of a Professor card.

The integer c is for the colour of the card.

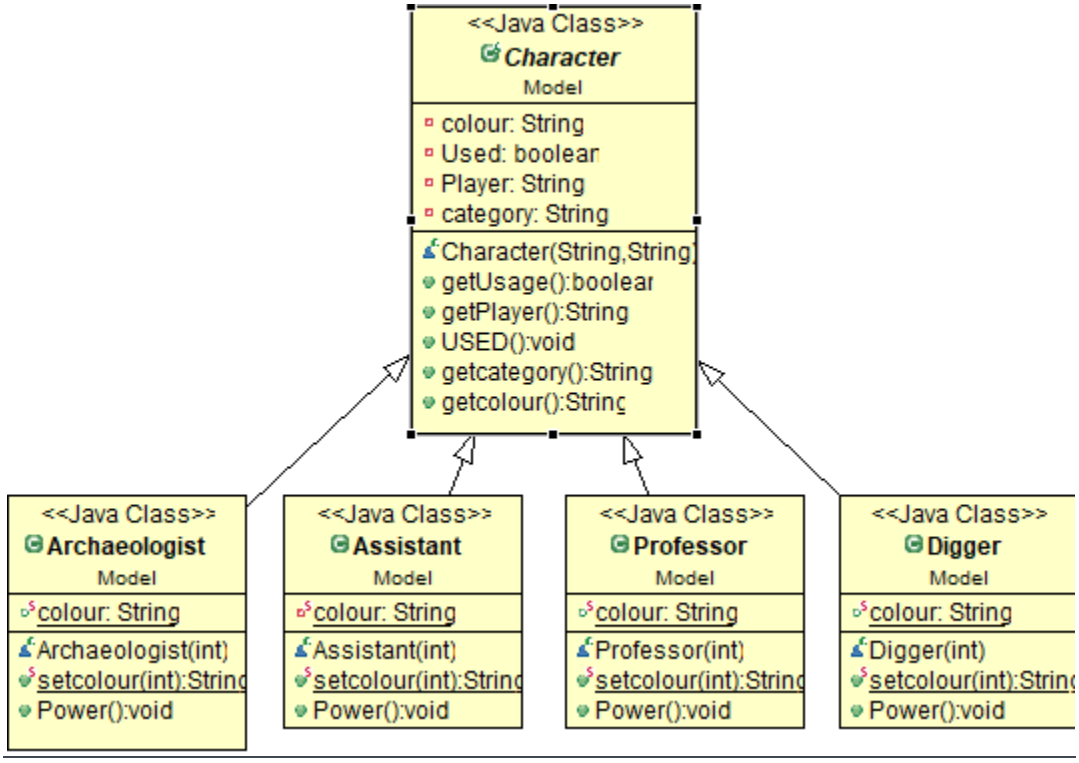
Methods:

1)public static String setcolour(int c);//Accessor(selector)

Sets the colour of a Professor card and returns it’s value.

2)public void Power()//Transformer

Sets the capabilities of the Professor card.



- **Class Player**

Η κλάση αυτή αναφέρεται στους παίκτες του παιχνιδιού. Αρχικοποιεί όλους τους παίκτες με τα σχετικα ID και ονόματά τους, καθώς και υπολογίζει τους πόντους κάθε παίκτη.

Ας τα δούμε πιο αναλυτικά:

Attributes:

1)private String name;//it's the name of the player.

2) private int points;//the points of a player.

3)protected static ArrayList<Character>character1;// The ArrayList for the characters of the 1st player.

4) protected static ArrayList<Character>character2;//The ArrayList for the characters of the 2nd player.

5) protected static ArrayList<Character>character3;//The ArrayList for the characters of the 3rd player.

6) protected static ArrayList<Character>character4;//The ArrayList for the characters of the 4th player.

7)private int ID;//the ID of a player

Constructor:
public Player(String name)
sets the name and the points of a player in the beginning of the game.

Methods:
1)public String getname(); Accessor
Returns the name of a player.

2)public int getpoints(); Accessor
Returns the number of points a player has.

3)public int getID();Accessor
Retuns the ID of a player. More specific returns the position of the player in the ArrayList of the Players. Also it starts from 0, for example the first player has ID 0 the second 1 etc.

4)public void get_characters(int c,Player p);//Transformer
Sets a pack of 4 characters with the same colour to each player. Each player will have different packs which will have different colours.

5)public void init_player(int id);//Transformer **(PhaseB)**
Initializes a player in the beginning of the game with his ID, in order to have easier access to that player later in the program.

6)public void setpoints(ArrayList<B_Position>Ptiles);//Transformer **(PhaseB)**
Sets the points of a player (at the end of the game) in order to let us know who is going to win the game.

- **Class Board**

Η κλάση Board είναι το ταμπλό του παιχνιδιού. Πάνω σε αυτό αρχικοποιούμε τα μέρη στα οποία τοποθετούμε τα tiles. Για αυτόν τον λόγο η κλάση αυτή περιέχει 5 ArrayLists τύπου B_Position (θα αναφέρουμε παρακάτω την ιδιότητα αυτής της κλάσης).

Στην φάση Β του πρότζεκτ η Board ήταν μία από τις συναρτήσεις που δεν πολυχρησιμοποιήθηκαν μιας και δεν χρειάστηκαν οι μέθοδοι που είχε (και έτσι διαγράφτηκαν). Στο μόνο πράγμα που χρησιμοποιείται η Board στην φάση Β είναι για να φτιαχτούν οι θέσεις των tiles στο ταμπλό καθώς και η κλήση της Bag ώστε να αρχικοποιηθούν τα Tiles.
Attributes:
PHASE A
1)private String category //category of the tile that the player chose to take from.

2)private boolean END = false;//checks if the game has finished or not.

3)protected ArrayList<B_Position>Mosaic = new ArrayList<B_Position>(); //The ArrayList for the MosaicTiles on the Board.

4) protected ArrayList<B_Position>Amphora = new ArrayList<B_Position>();//The ArrayList for the AmphoraTiles on the Board.

5) protected ArrayList<B_Position>Statues = new ArrayList<B_Position>();//The ArrayList for the StatueTiles on the Board.

6) protected ArrayList<B_Position>Skeletons = new ArrayList<B_Position>();//The ArrayList for the SkeletonTiles on the Board.

7) protected ArrayList<B_Position>Landslides = new ArrayList<B_Position>();//The ArrayList for the LandslideTiles on the Board.

8)private Bag bag = new Bag();//initialize the class Bag

PHASE B

1)public ArrayList<B_Position> Mosaic = new ArrayList<B_Position>();

2)public ArrayList<B_Position> Amphora = new ArrayList<B_Position>();

3) public ArrayList<B_Position> Sphinx = new ArrayList<B_Position>();

4) public ArrayList<B_Position> Caryatid = new ArrayList<B_Position>();

5) public ArrayList<B_Position> Skeletons = new ArrayList<B_Position>();

6) public ArrayList<B_Position> Landslides = new ArrayList<B_Position>();

Constructor:

public Board()

It doesn't do anything.

Methods:

PHASE A

1)public boolean END()//Observer

Checks if there are any LandslideTiles left in the bag and if there are not, then the boolean variable will be true and that means the end of the game.

2)public void setcategory(String category)//Transformer

Sets the category of the tile that a player chose to take from.

3)public String getcategory()//Accessor

Returns the category of the tile that a player chose to take from.

PHASE B

No methods.

- **Class B_Position**

Η κλάση B_Position αναφέρεται στα πλακίδια που βρίσκονται πάνω στο ταμπλό. Όπως είδαμε παραπάνω περιέχει 5 ArrayLists (ένα για κάθε είδος πλακίδιου)

Attributes:

1)private String category;//one of the 5 categories of the tiles on the board.

Constructors:

B_Position(String category)

Sets the value of the String category.

B_Position(String category,String colour)

Sets the value of the String category and the String colour and its only for Mosaic and Amphora Tiles.

B_Position(String category,String part,int k)

Sets the value of the String category and the String part. The integer k let us separate this constructor from the previous one. This Constructor is only for the SkeletonTiles.

Methods:

1)public String getcategory();//Accessor

Returns the category of a tile

2)public String getcolour();//Accessor

Returns the value of the String colour of a tile which is on the Board. Its referring to an AmphoraTile or a MosaicTile.

3)public String getpart();//Accessor

Returns the value of the String part of SkeletonTile which is on the Board.

- **Class Turn**

  Στην φάση Β η κλάση Turn διαγράφτηκε τελείως καθώς μου φάνηκε ευκολότερο και πρακτικότερο, τα γραφικά του παιχνιδιού καθώς και το turn κάθε παίκτη να γίνονται όλα στην κλάση View ώστε να μην χρειάζεται να καλούμε υπερβολικά πολλές μεθόδους από άλλες κλάσεις.

Η κλάση Turn αναφέρεται στον κάθε γύρο του παιχνιδιού και περιέχει μεθόδους που ελέγχουν την ροή του παιχνιδιού (διαγραφή ενός tile από την Bag αν επιλεχθεί από κάποιον παίκτη, εύρεση του παίκτη που έπαιξε τελευταία φορά, το ID (ο αριθμός του στην λίστα των παικτών) του παίκτη που παίζει, ελέχει αν ένας παίκτης τελείωσε τον γύρο του ή όχι, τοποθέτηση των tiles πάνω στο ταμπλό από την Bag(4 tiles κάθε φορά) και τοποθέτηση των tiles που βρίσκονται πάνω στο ταμπλό στον κάθε παίκτη ανάλογα με την επιλογή του).

Επίσης πρέπει να αναφέρω πως ένα turn τελειώνει όταν όλοι οι παίκτες έχουν παίξει μία φορά.

Ας την δούμε πιο αναλυτικά:

Attributes:

1)private int round_players;// The number of players who have not played in this turn.

2)private int last_player;//The last player who played in this turn.

3)private int num;//The number of players(in total).

4)private int currentID//The position of the player who is playing now in the ArrayList of the players.

5)private ArrayList<Player>player;//The ArrayList of the players.

6)Player p;//initialize the type of p in order to use methods from the class Player.

7)String name;//The name of the player who is playing now.

Constructor:

public Turn()

Initializes the variables: currentID, round_players, last_player and num.

Methods:

1)public void setID(Player ;p);//Transformer

sets the value of the variable currentID which is the ID of the player who is currently playing. The variable p is a pointer which shows who is the player we are looking for.

2)public int getID();//Accessor

Returns the value of the currentID, which is the ID of the player who is currently playing.

3)public boolean CheckIfPlayerFinished(Player p);//Observer

Returns true if a player has finished his turn else returns false.

4)public void NumberOfPlayers(int num);//Transformer

Sets the value of the variable num which represents the total number of players.

5)public int GetNumberOfPlayers();//Accessor

Returns the total number of players.

6)public void SetLastPlayer(int k);//Transformer

Sets the value of the variable last_player which is the posisiton in the ArrayList of Players of the most recent player who draw tiles from the bag.

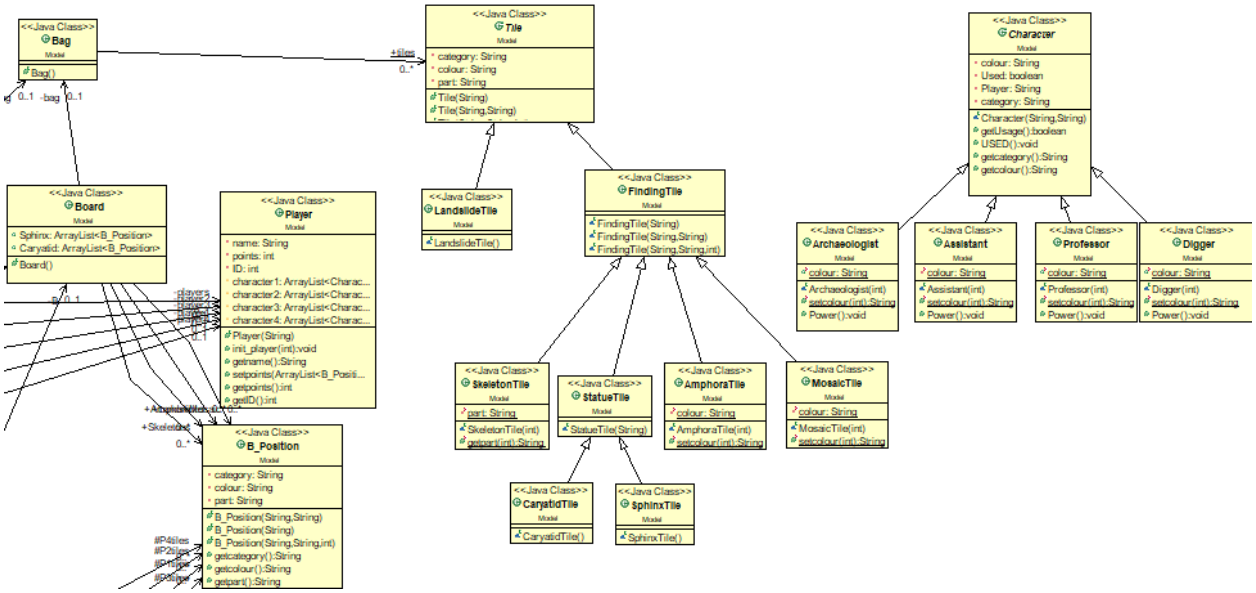
7)public int GetLastPlayer();//Accessor

Returns the value of the integer last_player

8)public void TakeTiles(Bag bag);//Transformer

First of all this method removes the 4 random tiles from the bag and puts them on the board. Then the player who is currently playing picks 2 of those tiles from the same category and adds them to his ArrayList.

That is all for the package model. The following picture shows what we have seen so far.



- Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Class Controller

Αυτή η κλάση είναι ουσιαστικά το μυαλό του παιχνιδιού. Είναι υπέυθυνη για τη δημιουργία ενός νέου παιχνιδιού, την δημιουργία παικτών(ονόματα) και φυσικά για την 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.

Attributes:

1)private Player player1,player2,player3,player4; //οι 4 παίκτες που θα παίξουν το παιχνίδι

2)private ArrayList<Player>players = new ArrayList<Player>();//Λίστα η οποία περιέχει τους παίκτες.

3)private Board board;

4)private Bag bag;

5)protected ArrayList<B_Position> P1tiles = new ArrayList<B_Position>(); //Player1 tiles
6)protected ArrayList<B_Position> P2tiles = new ArrayList<B_Position>(); //Player2 tiles
7)protected ArrayList<B_Position> P3tiles = new ArrayList<B_Position>(); //Player3 tiles
8)protected ArrayList<B_Position> P4tiles = new ArrayList<B_Position>(); //Player4 tiles

Constructor:

public Controller()
Initializes the Board and the 4 Players(sets their names) and starts the game.
Also initializes the IDES and the points of each player and puts all of them in an ArrayList in order to have access on them later on in the program.

Methods:

1)public Player getPlayers(int k);//Accessor

Maybe one of the most useful methods in the program. In the class View we will need this method a lot of times in order to give to the Players the tiles they are choosing.

- Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Class View

Η κλάση αυτή δημιουργεί ένα frame και μέσα σε αυτό ένα panel. Μέσα σε αυτό το panel υπάρχουν 5 panels για κάθε παίκτη, όπου στο 1ο περιλαμβάνονται οι κάρτες χαρακτήρων που έχει διαλέξει, στο 2ο τα tiles που διαθέτει, στο 3ο απεικονίζεται ποιος παίκτης έχει σειρά να παίξει, στο 4ο υπάρχει κουμπί για Draw Tiles και στο 5ο υπάρχει κουμπί για End Turn.

Επίσης υπάρχει ένα κεντρικό panel που είναι το ταμπλό του παιχνιδιού.Επιπλέον υπάρχουν κουμπιά για τα tiles του ταμπλού τα οποία ανάλογα από ποιόν παίκτη πατηθούν καταλήγουν στην κατοχή του και υπάρχουν και κουμπιά χαρακτήρων με συγκεκριμένες ιδιότητες.

ΣΗΜΑΝΤΙΚΟ ΣΧΟΛΙΟ

(Στην υλοποίηση μου δεν έχω φτιάξει κάτι για να περιορίζω πόσα tiles μπορεί να πάρει κάθε παίκτης στην σειρά του, όμως θεωρούμε πως παίζουμε δίκαια και παίρνουμε μόνο 2 tiles από το ταμπλό ανα γύρο από την ίδια κατηγορία. Επιπλέον δεν πρέπει να ξαναπατάμε τα tiles που έχει ο κάθε παίκτης στην διάθεσή του, καθώς δεν τα κάνω disable λόγω του ότι μετά είναι δύσκολο να διακριθούν).

Σε γενικές γραμμές η κλάση View δίνει τις κατάλληλες εντολές με βάση τον παίκτη και το πρόγραμμα τις υλοποιεί και με γραφικό τρόπο αλλά και με πρακτικό(γίνονται σωστά οι συνδέσεις με τους παίκτες,,τα tiles και τα ArrayLists τους).

Ας δούμε τώρα την κλάση αυτή πιο αναλυτικά:

Constructor

Calls the method initialize()

Methods:

- Private void Initialize()//Transformer
  Initializes the frame. For example sets the coordinates of each panel and Label, sets the image of the board, adds a panel into a list of panels etc.
- Public static void fill(int[ ]j,int num)//Transformer
  Fills the array j with 0 which helps in the position of the tiles on the board(one after another).
- Public void switchE_T(JButton E_T);//Transformer

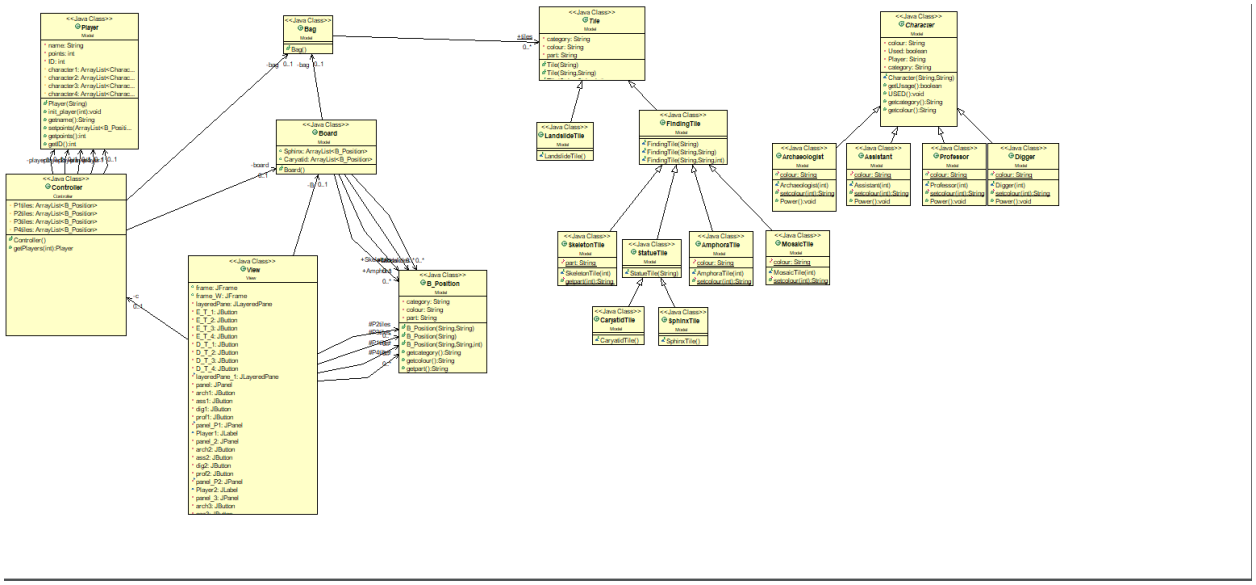
Changes the panel of the Button End Turn in order to let the players press it,since every turn only one End_Turn button is enabled (for safety reasons).

- Public void switchD_T(JButton D_T);//Transformer
  Changes the panel of the Button Draw Tiles in order to let the players press it,since every turn only one Draw_Tiles button is enabled (for safety reasons).
- Public void switchPanels(JPanel panel);//Transformer
  Changes the panel of the characters for every player
- Public void switchPlayer(JPanel panel_P);//Transformer
  Changes the panel which shows the Labels of players.
- Public void switchTiles(JPanel panel_T);//Transformer
  Changes the panel of each player's tiles.

- The following methods are transformers which are used only when a tile button is pressed. They transfer this button to the panel of the player who pressed it.
  - Public void MosaicTileP(int q,JButton MosaicTile,B_Position BP);
  - Public void SkeletonTileP(int q,JButton MosaicTile,B_Position BP);
  - Public void AmphoraTileP(int q,JButton MosaicTile,B_Position BP);
  - Public void SphinxTileP(int q,JButton MosaicTile,B_Position BP);
  - Public void CaryatidTileP(int q,JButton MosaicTile,B_Position BP);
- DrawTiles(int q)//Transformer
  - One of the most important methods in the program. Whenever a Draw Tiles button is pressed a function which is able to identify who is the player who pressed it, calls this function. This function calls all the previous methods in order to place each button in the right position on the board. Also in each category of a button there is another method, named ActionListener, which is called whenever someone presses the button it represents.
    When this method is called, it calls the previous method according to the category of the pressed tile.
    Also this method checks if the game is going to end, by announcing the Winner of the game.
- The following methods are also transformers which are used in order to set all the panels and the buttons for the player who has the turn to play, which means that every button of that player must be enabled and all the other buttons that are not connecting to that player must be disabled. Also with these methods we basically make the turn of a player. By the end of each of these methods there is an ActionListener of a button type End Turn. If this button is pressed then the player who is playing right now passes his turn to the next player (the player1 to the player2, the player2 to player3 etc.).Lastly when the player4 finished his turn and presses this button, the turn passes to the player1 and the program repeats the same process until the LandslideTiles on the board are 16 (that's when the program finishes).
  - Private void action1();

    - Private void action2();
    - Private void action3();
    - Private void action4();
- Last but not least we have 4 more Transformers which are used only one time in the program. They are part of the initialize and they represent the initialization of each player and the starting of the game.
    - Private void player1();
    - Private void player2();
    - Private void player3();
    - Private void player4();

- Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



Το παραπάνω διάγραμμα απεικονίζει τις συνδέσεις όλων των κλάσεων που αναφέραμε παραπάνω. Όπως παρατηρούμε όλα συνδέονται μεταξύ τους εκτός από την κλάση Character η οποία υπάρχει απλά και μόνο για να αρχικοποιεί τις κάρτες χαρακτήρα. Όπως βλέπουμε και στο διάγραμμα το μοίρασμα των χαρακτήρων στους παίκτες γίνεται στην κλάση Player χρησιμοποιώντας ένα ArrayList τύπου Character. Δηλαδή η κλάση Character δεν χρειάζεται να συνδέεται με καμία αλλη κλάση παρά μόνο με τις υποκλάσεις της.
Όσων αφορά την Controller, βλέπουμε ότι συνδέεται με τις κλάσεις Board,Player, Bag οι οποίες με την σειρά τους συνδέονται με τις υπόλοιπες κλάσεις. Δηλαδή για να λειτουργήσουν όλες οι κλάσεις αρκεί να τρέξουμε την κλάση Controller.
Έπειτα βλέπουμε την κλάση Tile η οποία συνδέεται με την κλάση Bag και της δίνει τα tiles της ώστε να τα αποθηκεύσει σε ενα ArrayList τύπου tile. Όσων αφορά την κλάση B_Position, ελέγχονται(παίρνουν τις τιμές τους) από την κλάση Board και Player αντίστοιχα και προσφέρουν και οι 2 μεθόδους χρήσιμες για την ομαλή λειτουργία του παχνιδιού.

Τέλος βλέπουμε πως η κλάση View συνδέεται με την Controller, την B_Position και την Board. Η View δίνει τιμές(επιλογές του χρήστη) στις παραπάνω κλάσεις και με βάση τις επιλογές των παικτών, δημιουργείται το παιχνίδι.

<u>Λειτουργικότητα:</u>
Στην Β φάση του πρότζεκτ τα πράγματα που δεν κατάφερα είναι σχετικά λίγα.
Αρχικά έχω κάνει αρχικοποίηση των παικτών, των tiles στο Bag και στις κάρτες χαρακτήρα.

Έχω κρατήσει τήρηση σειράς (παίκτης 1 έως 4), καθώς και υλοποιώ ορθά το τράβηγμα των πλακιδίων από την σακούλα και τα τοποθετώ στις αντίστοιχες περιοχές τους στο ταμπλο.

Υλοποιώ σωστά το τράβηγμα των πλακιδίων από το ταμπλό. Σε αυτό εδώ μόνο το κομμάτι πρέπει να ξαναναφέρω πως δεν έχω βάλει περιορισμό στο πόσα πλακίδια μπορεί κάποιος παίκτης να πάρει και από που, επειδή την ώρα που έφτιαχνα το πρόγραμμα θεώρησα πως θα παίζαμε δίκαια με τους κανόνες του παιχνιδιού.

Ελέγχω σωστά την περιοχή κατολίσθησης και το παιχνίδι τελειώνει όταν γεμίσει με 16 LandSlideTiles το μεσαίο τετραγωνάκι.

Υπολογίζω ορθά τους πόντους για τα μωσαικά,σκελετούς και από τους αμφορείς δεν έχω πάρει μόνο την περίπτωση που έχουμε μόνο 3 συνδυσμούς.

Έχω φτιάξει τις κάρτες χαρακτήρα και τις έχω εκχωρήσει σε κάθε παίκτη καθώς και στα αντίστοιχα panels τους. Παρ'όλα αυτά ενώ έχω φτιάξει τα κουμπιά τους και τα έχω βάλει να γίνονται disable όταν πατιούνται δεν έχω βάλει να κάνουν κάτι.

Άρα συνοψίζοντας αυτά που δεν έχω καταφέρει να κάνω στο πρόγραμμα είναι ο υπολογισμός πόντων για αγάλματα καθώς και τις ιδιότητες των χαρακτήρων.

Συμπεράσματα:

Η υλοποίηση αυτού του πρότζεκτ ήταν μία ωραία και καινούρια εμπειρία η οποία μας έβαλε λίγο πιο βαθιά στον προγραμματισμό. Εμένα προσωπικά με βοήθησε αρκετά να καταλάβω πως δουλεύουν τα γραφικά και πως πρέπει να χρησιμοποιούμε περισσότερες μεθόδους για την υλοποίηση μεγάλων προβλημάτων. Με την τελευταία φράση αναφέρομαι στην κλάση View η οποία μου πήρε πολύ χρόνο να την υλοποιήσω λόγω του ότι έπρεπε να πάρω κάθε περίπτωση και να βάλω τους κατάλληλους ελέγχους. Κατα τ'άλλα δεν μπορώ να πω ότι με δυσκόλεψε κάτι άλλο στην εργασία παρά μόνο η αρχική σκέψη για την υλοποίησή της, καθώς δεν ήξερα από που να ξεκινήσω. Ήταν μία πολύ εκπαιδευτική εργασία η οποία κατάφερε να μου κινήσει το ενδαφέρον για το μάθημα και γενικά για την java ως γλώσσα προγραμματισμού. Τέλος πρέπει να αναφέρω πως υπήρχε μεγάλη διαφορά μεταξύ της 1ης φάσης του πρότζεκτ με την 2η όσων αφορά τις μεθόδους που χρησιμοποίησα. Στην 2η φάση αναγκάστηκα να σβήσω τις περισσότερες μεθόδους από πολλές κλάσεις, εφόσον δεν χρησιμοποιούνταν για την υλοποίηση του προγράμματος. Επιπλέον στην 2η φάση του πρότζεκτ η κύρια κλάση (η πιο σημαντική) είναι η View μιας και δεν απεικονίζει μόνο γραφικά αλλά γεμίζει τις λίστες (παικτών, tiles,board), τηρεί σειρά και δημιουργεί την αλληλεπίδραση μεταξύ ταμπλού, καρτών χαρακτήρων και παικτών.

ΤΕΛΟΣ