



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

OBJECT-ORIENTED  
PROGRAMMING HY-  
PROJECT 2020-2021

Νικόλα Κοβάτσεβιτς

4285

12/03/2020

## Περιεχόμενα

1. Εισαγωγή.....	1
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model.....	1
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller.....	13
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	15
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....	15

## 1. Εισαγωγή

Η υλοποίηση της εργασίας βασίστηκε πάνω στο μοντέλο MVC(MODEL – VIEW – CONTROLLER). Ο σκοπός είναι ο CONTROLLER να είναι η κύρια σύνδεση μεταξύ του MODEL και του VIEW. Οπότε στην συνέχεια θα αναφερθούμε κυρίως στο MODEL και το CONTROLLER, και στο τέλος στο VIEW το οποίο δεν είναι τόσο σημαντικό σε αυτή την φάση, και για τον λόγο αυτό θα αναφερθεί πιο συνοπτικά.

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

Σε αυτό το πακέτο περιέχονται η διεπαφή Tile, οι κλάσεις LandslideTile και FindingTile, καθώς και οι κλάσεις που κληρονομούν την κλάση FindingTile, η διεπαφή Character και οι κλάσεις που την κληρονομούν και οι κλάσεις Player, GraveRobber, Tile Bag και Board

### Tile Interface and Classes of Tiles

Με την διεπαφή Tile μπορούμε να προσπελάσουμε τα δεδομένα χωρίς να χρειάζεται να ορίσουμε αν ένα tile είναι απλό η LandslideTile.

#### Το Interface αυτό μας παρέχει τις εξής μεθόδους

1. void setColor(String color); Transformer(Mutative)

Sets the color of a tile.

2. String getColor(); Accessor(Selector)

Returns the color of the tile.

3. void setType(String type); Transformer(Mutative)

Sets the type of a tile.

4. String getType(); Accessor(Selector)

Returns the type of the tile.

Στην συνέχεια έχουμε την FindingTile και την LandslideTile που υλοποιούν την Tile.

### Class FindingTile

Εδώ θα αναφέρουμε τα attributes και τις υπόλοιπες μεθόδους της κλάσης.

### Τα attributes

1. `protected String Color; // the color of the tile`
2. `protected String Type; // the type of the tile`

### Οι μέθοδοι

1. `public void setColor(String color); Transformer(Mutative)`  
Sets the color of a tile.
2. `public String getColor(); Accessor(Selector)`  
Returns the color of the tile.
3. `public void setType(String type); Transformer(Mutative)`  
Sets the type of a tile.
4. `public String getType(); Accessor(Selector)`  
Returns the type of the tile.

### Class **LandslideTile**

Εδώ θα αναφέρουμε τα attributes και τις υπόλοιπες μεθόδους της κλάσης.

### Τα attributes

1. `private String Color; // the color of the tile`
2. `private String Type; // the type of the tile`

### Οι μέθοδοι

1. `public final void setColor(String color); Transformer(Mutative)`  
Sets the color of the tile to clear.
2. `public String getColor(); Accessor(Selector)`  
Returns the color of the tile.
3. `public final void setType(String type); Transformer(Mutative)`  
Sets the type of the tile to LandslideTile
4. `public String getType(); Accessor(Selector)`  
Returns the type of the tile.

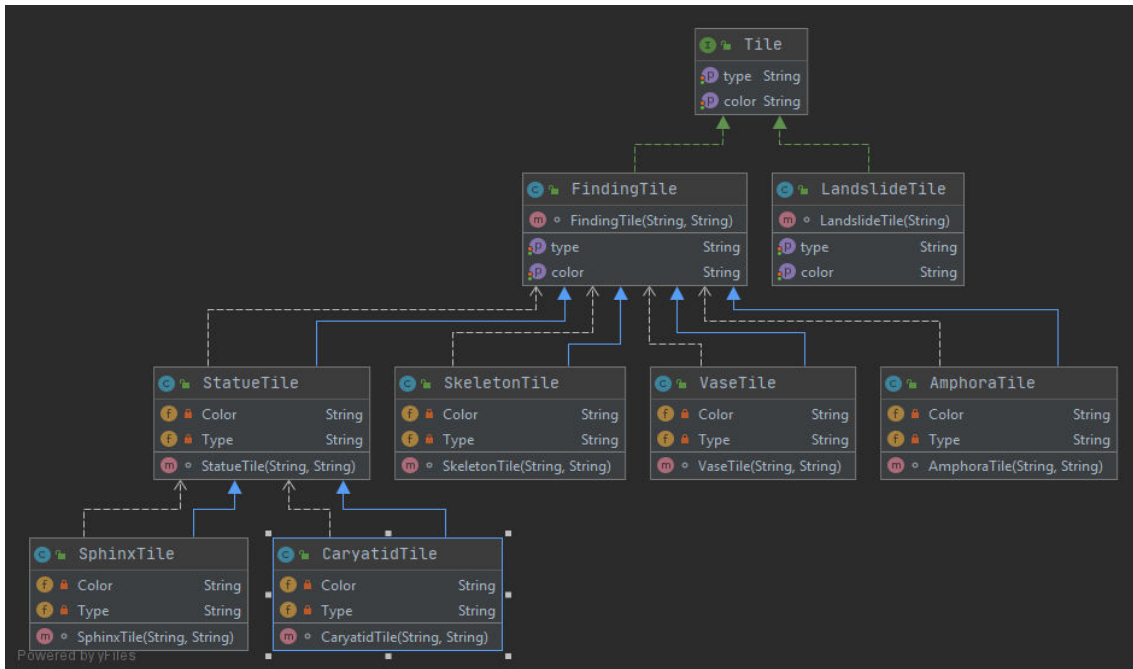
## Classes AmphoraTile, VaseTile, SkeletonTile, StatueTile

Αυτές οι κλάσεις κάνουν extend την FindingTile και μέσω της εντολής super, αποκτούν πρόσβαση σε αυτήν και αρχικοποιούν τα πεδία Color και Type.

## Classes CaryatidTile και SphinxTile

Αυτές οι κλάσεις κάνουν extend την StatueTile και μέσω της εντολής super, αποκτούν πρόσβαση σε αυτήν και αρχικοποιούν τα πεδία Color και Type.

Τέλος εδώ θα δείξουμε μια αναπαράσταση των class που έχουν σχέση με τα tiles



μέσω UML.

## Characters Interface and Characters Classes

Με την διεπαφή Characters μπορούμε να προσπελάσουμε τα δεδομένα χωρίς να χρειάζεται να ορίσουμε τι character είναι η κάθε character card.

### Το interface αυτό μας παρέχει τις εξής μεθόδους

1. String getCharacter(); Accessor(Selector)  
Returns the character of the card.
2. void setPlayer(String Player); Transformer(Mutative)

Sets the player to which the card belongs to.

3. `.String getPlayer(); Accessor(Selector)`

Returns the name of the owner of the card.

4. `void setUsed(boolean used); Transformer(Mutative)`

Sets the if the card is used.

5. `boolean getUsed(); Observer`

Returns if the card is used.

6. `void setColor(String color); Transformer(Mutative)`

Sets the color of the card.

7. `String getColor(); Accessor(Selector)`

Returns the color of the card.

8. `Void use(String area); Transformer(Mutative)`

Uses the card.

## **Classes Archeologist, Assistant, Digger and Professor**

Οι κλάσεις αυτές υλοποιούν την διεπαφή Characters.

Εδώ αναφέρουμε τα attributes και τις μεθόδους των κλάσεων αυτών.

### **Τα attributes**

1. `final String Character; // the character of the card. Same with the name of the  
// class`
2. `private String player; // the player that the card belongs to`
3. `private boolean Used; // shows if the card is used`
4. `private String color; // the color of the card`

### **Οι μέθοδοι**

1. `String getCharacter(); Accessor(Selector)`  
Returns the character of the card.
2. `void setPlayer(String Player); Transformer(Mutative)`  
Sets the player to which the card belongs to.

3. `.String getPlayer(); Accessor(Selector)`

Returns the name of the owner of the card.

4. `void setUsed(boolean used); Transformer(Mutative)`

Sets the if the card is used.

5. `boolean getUsed(); Observer`

Returns if the card is used.

6. `void setColor(String color); Transformer(Mutative)`

Sets the color of the card.

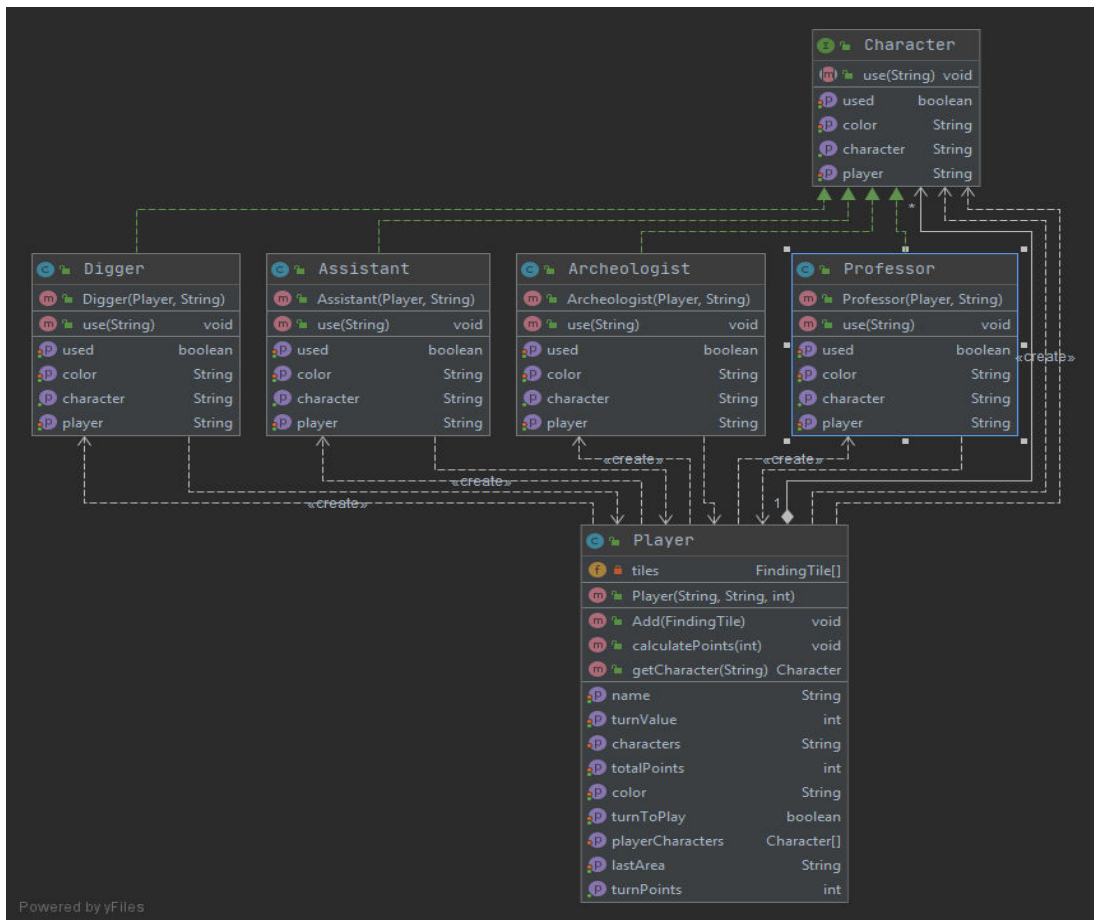
7. `String getColor(); Accessor(Selector)`

Returns the color of the card.

8. `Void use(String area); Transformer(Mutative)`

Uses the card.

Τέλος εδώ θα δείξουμε μια αναπαράσταση των class που έχουν σχέση με τα tiles μέσω UML.



## Class Player

Η κλάση αυτή αναπαριστά τον παίκτη μέσα στο παιχνίδι.

### Ta attributes

1. `private String Name; // the name of the player`
2. `private FindingTile[] tiles = new FindingTile[111] // the area where the player  
// stores his tiles.`
3. `private String area; // the last area where the player took tiles from.`
4. `private int TotalPoints; // the total points of the player`
5. `private int turnPoints; // the points the player collects every turn.`
6. `private int turnValue; // the number of the current player's turn.`
7. `private String Color; // the color of the player.`
8. `Private Character[] playerCharacters; // the character cards of the player`



9. `private boolean turnToPlay; // if it's the player's turn to play.`

### Οι μέθοδοι

1. `public void setName(String Name); Transformer(Mutator)`  
Sets the name of the player.
2. `public String getName(); Accessor(Selector)`  
Returns the name of the player.
3. `public void add(FindingTile Tile); Transformer(Mutative)`  
Adds a tile to the player's tiles.
4. `public void setLastArea(String area); Transformer(Mutative)`  
Sets the last area from which the player took a tile.
5. `public String getLastArea(); Accessor(Selector)`  
Returns the name of the last area from which the player took a tile.
6. `public void calculatePoints(int points); Transformer(Mutative)`  
Calculates the points the player got this turn.
7. `public int getTurnPoints(); Accessor(Selector)`  
Returns the points the player got this turn.
8. `public void setTotalPoints(int totalPoints); Transformer(Mutative)`  
Calculates the total points the player got in this game.
9. `public int getTotalPoints(); Accessor(Selector)`  
Returns the total points of the player.
10. `public void setTurnValue(int turnValue); Transformer(Mutative)`  
Sets the turn number that this player plays.
11. `public int getTurnValue(); Accessor(Selector)`  
Returns the turn number.
12. `public void setColor(String color); Transformer(Mutative)`  
Sets the color of the player.
13. `public String getColor(); Accessor(Selector)`

Returns the color of the player.

14. `public void setTurnToPlay(boolean turnToPlay); Transformer(Mutative)`

Sets if this is the turn for this player to play.

15. `public boolean isTurnToPlay(); Observer`

Returns if this is the turn for the player to play.

16. `public void setPlayerCharacters(Character[] playerCharacters);`

`Transformer(Mutative)`

Sets the cards of the player.

17. `public void setCharacters(string Color); Transformer(Mutative)`

Adds the cards of a certain color to the player's cards.

18. `public Character getCharacter(String CharName); Accessor(Selector)`

Returns a certain player's card.

## Class GraveRobber

Αυτή η κλάση θα χρησιμοποιηθεί μόνο σε περίπτωση που το παιχνίδι θα παιχτεί με 1 παίκτη αντί για 4. Ο GraveRobber αναλαμβάνει να παίρνει όλα τα tiles από το ταμπλό στο τέλος κάθε γύρου.

### Ta attributes

1. `private String Name; // the name of the GraveRobber.`
2. `private int totalPoints; // the total points of the GraveRobber.`

### Οι μεθόδους

1. `public void getTiles(Board board, String area); Transformer(Mutative)`  
Removes all the tiles from the board except for the Landslide Tiles.
2. `Public void setTotalPoints(int totalPoints); Transformer(Mutative)`  
Sets the total points of the GraveRobber.
3. `public int getTotalPoints(); Accessor(Selector)`

Returns the total points of the GraveRobber.

## **Class TileBag**

Αυτή η κλάση υλοποιεί την σακούλα που περιέχει τα tiles τα οποία τραβούν οι players σε κάθε γύρο.

### **Ta attributes**

1. private Tile[] tiles; // the space where the tiles are stored inside the bag.
2. private int TilesLeft; // the number of tiles left inside the bag.

### **Οι μέθοδοι**

1. public void initializeBag(); Transformer(Mutative)  
Initializes the bag filling it with tiles.
2. public void shuffleBag(); Transformer(Mutative)  
Shuffles the tiles inside the bag.
3. public void Draw(); Transformer(Mutative)  
Draws a tile from the bag.
4. public void setTilesLeft(int tilesLeft); Transformer(Mutative)  
Sets the number of the tiles that are left in the bag.
5. public int getTilesLeft(); Accessor(Selector)  
Returns the number of tiles left in the bag.

## **Class Board**

Η κλάση αυτή υλοποιεί το ταμπλό του παιχνιδιού, καθώς και ενέργειες που γίνονται σε αυτό.

### **Ta attributes**

1. private LanslideTile[] LandArea; //the area where you place the Landslide  
//Tiles
2. private AmphoraTile[] AmphoraArea; //the area where you place the Amphora  
//Tiles
3. private StatueTile[] StatueArea; //the area where you place the Statue  
//Tiles

4. private VaseTile[] VaseArea; //the area where you place the Vase  
//Tiles
5. private SkeletonTile[] SkeletonArea; //the area where you place the Skeleton  
//Tiles
6. private TileBag Bag; // the bag from which the players draw the tiles.

### Οι μέθοδοι

1. public void Start(); Transformer(Mutative)  
initializes the areas.
2. public void setAmphoraArea(AmphoraTile[] amphoraArea);  
Transformer(Mutative)  
Sets the Amphora area.
3. public AmphoraTile[] getAmphoraArea(); Accessor(Selector)  
Returns the Amphora area.
4. public void setLandslideArea(LandslideTile[] landslideArea);  
Transformer(Mutative)  
Sets the Landslide area.
5. public LandslideTile[] getLandslideArea(); Accessor(Selector)  
Returns the Landslide area.
6. public void setVaseArea(VaseTile[] vaseArea);  
Transformer(Mutative)  
Sets the Vase area.
7. public VaseTile[] getVaseArea(); Accessor(Selector)  
Returns the Vase area.
8. public void setStatueArea(StatueTile[] statueArea);  
Transformer(Mutative)  
Sets the Statue area.
9. public StatueTile[] getStatueArea(); Accessor(Selector)  
Returns the Statue area.

10. `public void setSkeletonArea(SkeletonTile[] skeletonArea);`  
`Transformer(Mutative)`  
 Sets the Skeleton area.
11. `public SkeletonTile[] getSkeletonArea();` `Accessor(Selector)`  
 Returns the Skeleton area.
12. `public void setBag(TileBag bag);` `Transformer(Mutative)`  
 Sets the Bag with the tiles.
13. `public TileBag getBag();` `Accessor(Selector)`  
 Returns the Bag with the tiles.
14. `public void removeTile(String area);` `Transformer(Mutative)`  
 Removes a tile from a specific area.
15. `public void removeAllTilesArea(String area);` `Transformer(Mutative)`  
 Removes all the tiles from an area.
16. `Public void insertTiles(int tile, String Area);` `Transformer(Mutative)`  
 Inserts a number of tiles in a given area.

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

Η class Controller είναι το κέντρο ελέγχου του παιχνιδιού. Η κλάση αυτή είναι υπεύθυνη για την δημιουργία ενός καινούριου παιχνιδιού, την δημιουργία στιγμιτύπων παικτών και γύρων, καθώς και για την σύνδεση μεταξύ του πακέτου View και του πακέτου Model. Η συγκεκριμένη κλάση παίρνει επίσης τις επιλογές των παικτών μέσα από το γραφικό περιβάλλον του View, και ενεργεί κατάλληλα ώστε το παιχνίδι να παίζεται σωστά. Τέλος ο Controller μετράει το σκορ, και κανονίζει πότε το παιχνίδι τελειώνει.

#### Τα attributes της κλάσης

1. `private int playerNumber;` // the number of the players.
2. `private Board b;` // the board of the game.
3. `private PlayerMenu p;` // the menu from where the number of players is chosen.

4. `private PlayerInfo I; // the menu from where the attributes of each player are  
//chosen.`
5. `private Player[] players; // the players of the game.`
6. `private String Winner; // the name of the winner of the game.`
7. `Private boolean gameOver; // if the game is over.`
8. `Private boolean not_started; // if the game has not started yet.`
9. `Private int j = 0; // a counter for the player number`  

\*We need the counter j to show the player info menu j times j = player number.

### Οι μέθοδοι

1. `public void setPlayerNumber(int playerNumber); Transformer(Mutative)`  

Sets the number of the players.
2. `Public int getPlayerNumber(); Accessor(Selector)`  

Returns the number of the players;
3. `public void initializeTurn(); Transformer(Mutative)`  

Initializes the first turn.
4. `Public void NewTurn(Player player, int turnValue); Transformer(Mutative)`  

Sets the new turn to the right player.
5. `Public void turnEnd(Player player); Transformer(Mutative)`  

Ends the turn for a player.
6. `Public void setGameOver(boolean gameOver); Transformer(Mutative)`  

Sets if the game is over.
7. `Public boolean getGameOver(); Observer`  

Returns if the game is over.
8. `Public void addPlayer(String Color, String Name); Transformer(Mutative)`  

Adds a player with given attributes to the game.
9. `Public Player getPlayer(int playerNumber); Accessor(Selector)`

Returns a specific player.

10. Public void setPlayers(int players); Transformer(Mutative)

Sets the players of the game according to the players number.

11. Public void setNot\_Started(boolean not\_Started); Transformer(Mutative)

Sets if the game has not started yet.

12. Public boolean isNot\_Started(); Observer

Returns if the game has not started.

13. Public int getPlayerTurnPoints(Player player); Accessor(Selector)

Returns the points a player got this turn.

14. Public int getTotalPlayerPoints(Player player); Accessor(Selector)

Returns the total points of a player.

15. Public void getStartingTiles(Player player, TileBag bag);  
Transformer(Mutative)

Gives to a player his 4 starting tiles.

16. Public void playTurn(Player player); Transformer(Mutative)

Plays the player's turn.

17. Public void setWinner(Player player); Transformer(Mutative)

Sets one player as the winner of the game.

18. Public void findWinner(); Transformer(Mutative)

Calculates the points to declare the winner.

19. Public String getWinner(); Accessor(Selector)

Returns the name of the winner of the game.

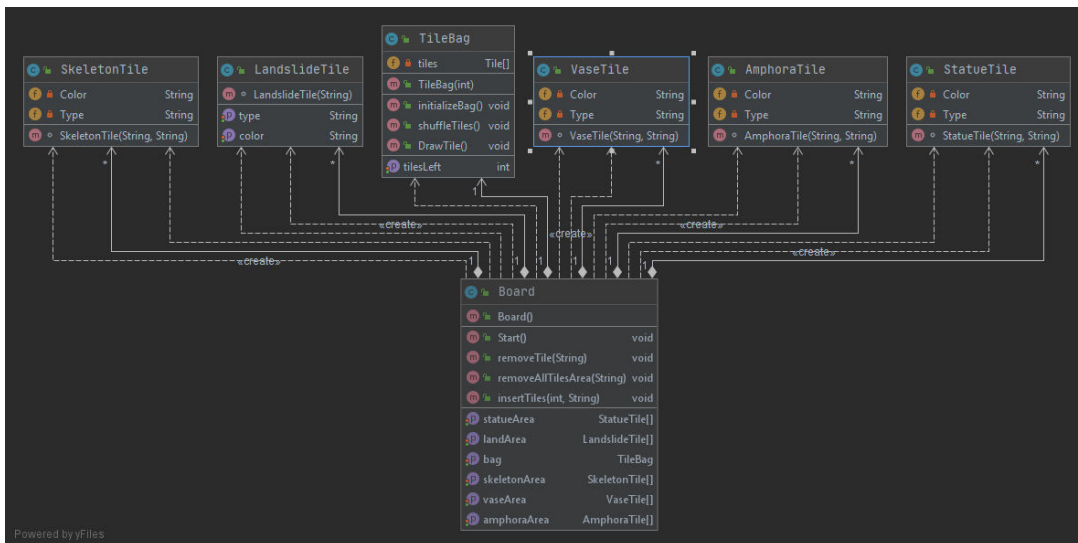
#### 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Αυτό το πακέτο θα αποτελείται από τρεις κλάσεις. Μια για το μενού των παικτών, μια για το μενού των attributes των παικτών, και μια για το παιχνίδι. Οι κλάσεις για τα μενού θα αποτελούνται κυρίως από ένα frame και κουμπιά/textfields για τις επιλογές του παίκτη(τον αριθμό των παικτών και το χρώμα, όνομα του κάθε παίκτη αντίστοιχα). Η

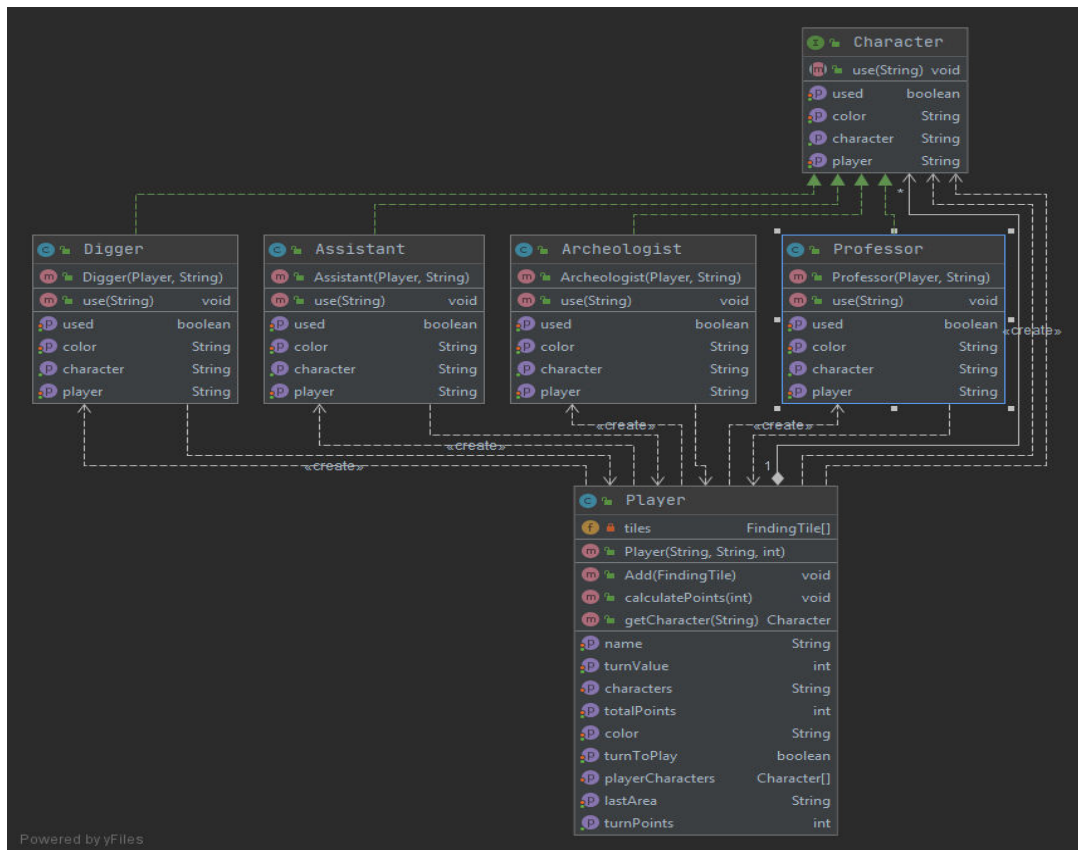
κλάση του παιχνιδιού, θα αποτελείται από ένα frame, ένα κύριο panel το οποίο θα περιεχέι το ταμπλό, τα tiles του κάθε παίκτη, τις κάρτες του κάθε παίκτη, τα κουμπιά για Draw και Turn End, τα στοιχεία του παίκτη(textfields), ένα panel με image για το background, 5 panels για τις areas, κουμπιά για τα tiles(131 ή 127 ανάλογα με τον αριθμό των παικτών) και τις κάρτες(16), και τέλος κουμπιά για New Game, Save Game, Continue και Exit(alternatively θα μπορούσα να φτιάξω ένα ακόμα menu για αυτές τις λειτουργίες, και να έχω απλά ένα κουμπί για να καλεί το menu). Σε κάθε γύρο ο παίκτης που είναι η σειρά του θα πατάει το κουμπί Draw και θα τραβάει 4 tiles(Εκτός από τον πρώτο του γύρο που θα παίρνει αυτόματα 4 tiles) και μετά θα τα τραβάει στην περιοχή τους. Επειτα θα παίρνει 2 tiles από όποια περιοχή θέλει πατώντας τα. Τέλος θα πατάει τις κάρτες εάν το επιθυμεί και θα πατάει το κουμπί End Turn για να τελειώσει τον γύρο του. Στο τέλος του παιχνιδιού, το σκορ θα μετριέται και θα ανακοινώνεται ο νικητής, μεσώ panel και textfield.

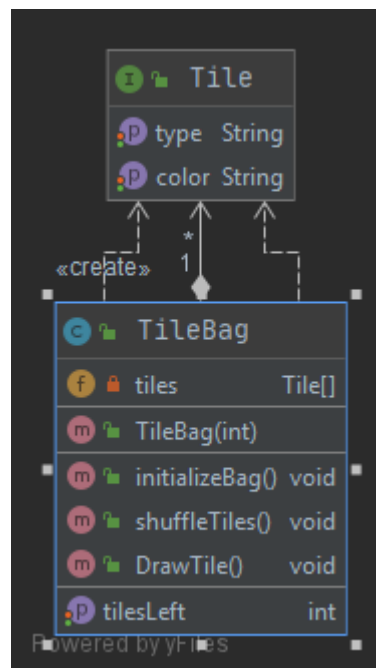
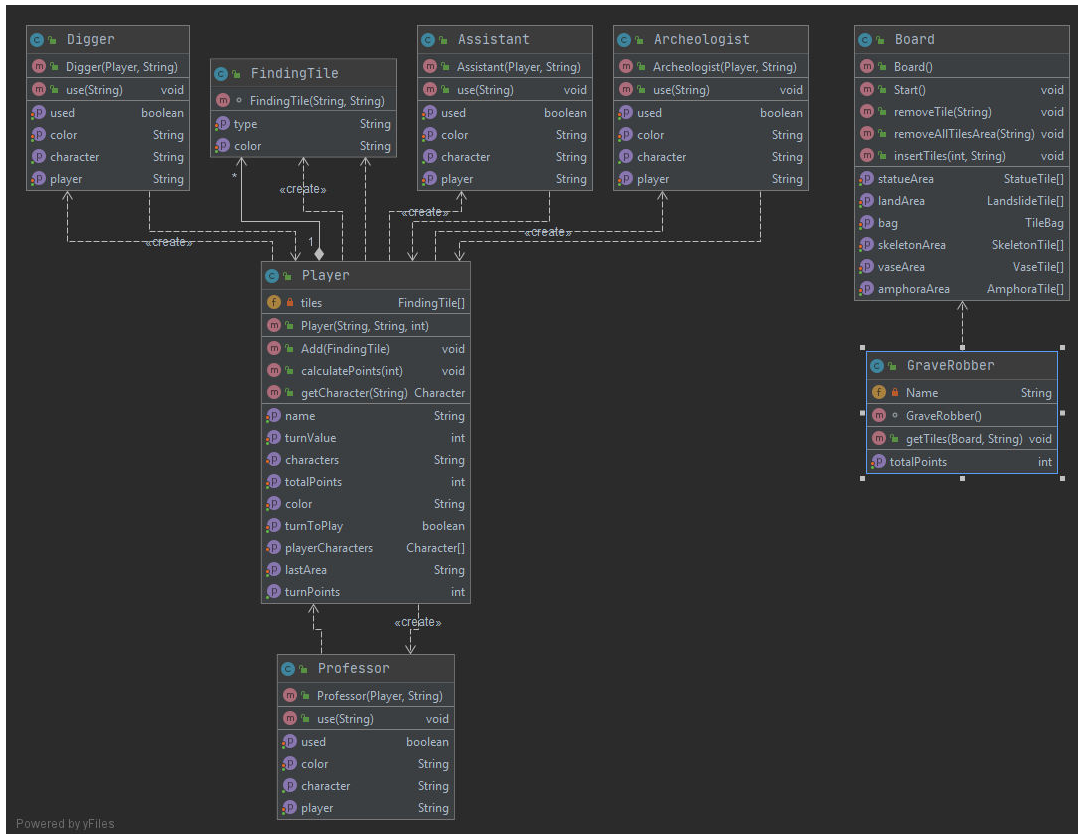
## 5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

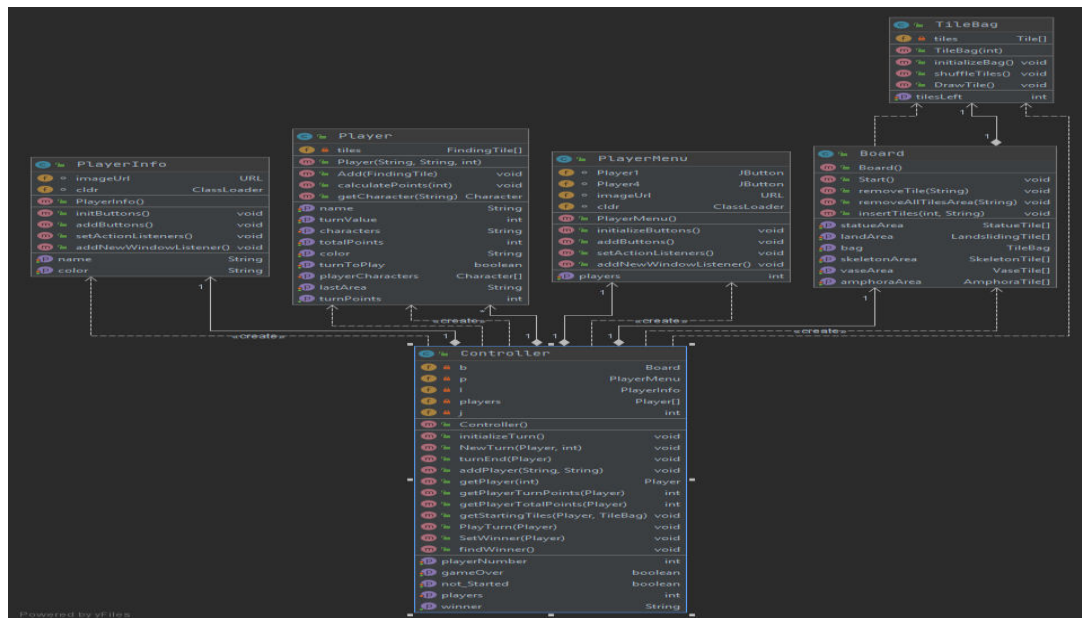
Στην συνέχεια θα συμπεριληφθούν τα διαγράμματα UML για την αλληλεπίδραση μεταξύ κλάσεων και πακέτων.

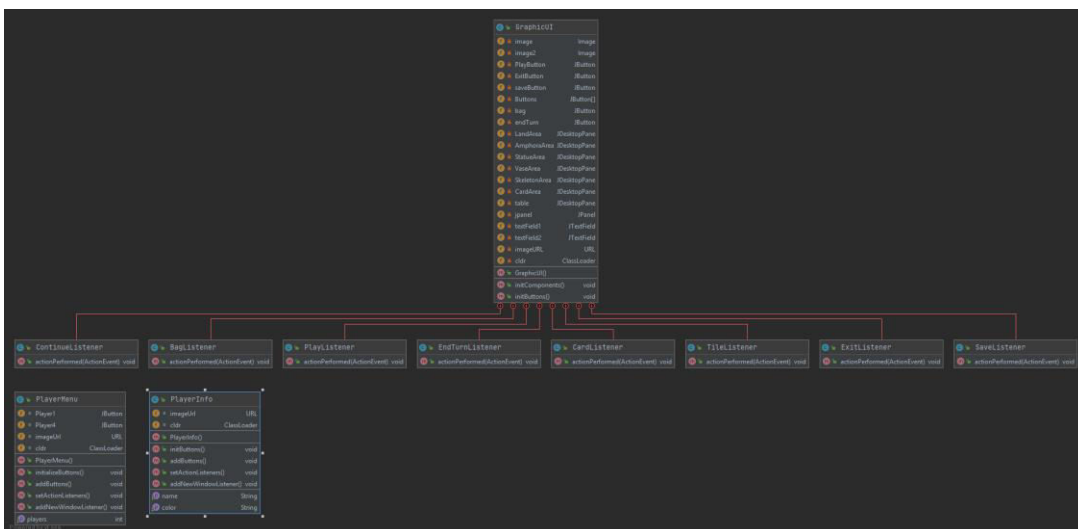
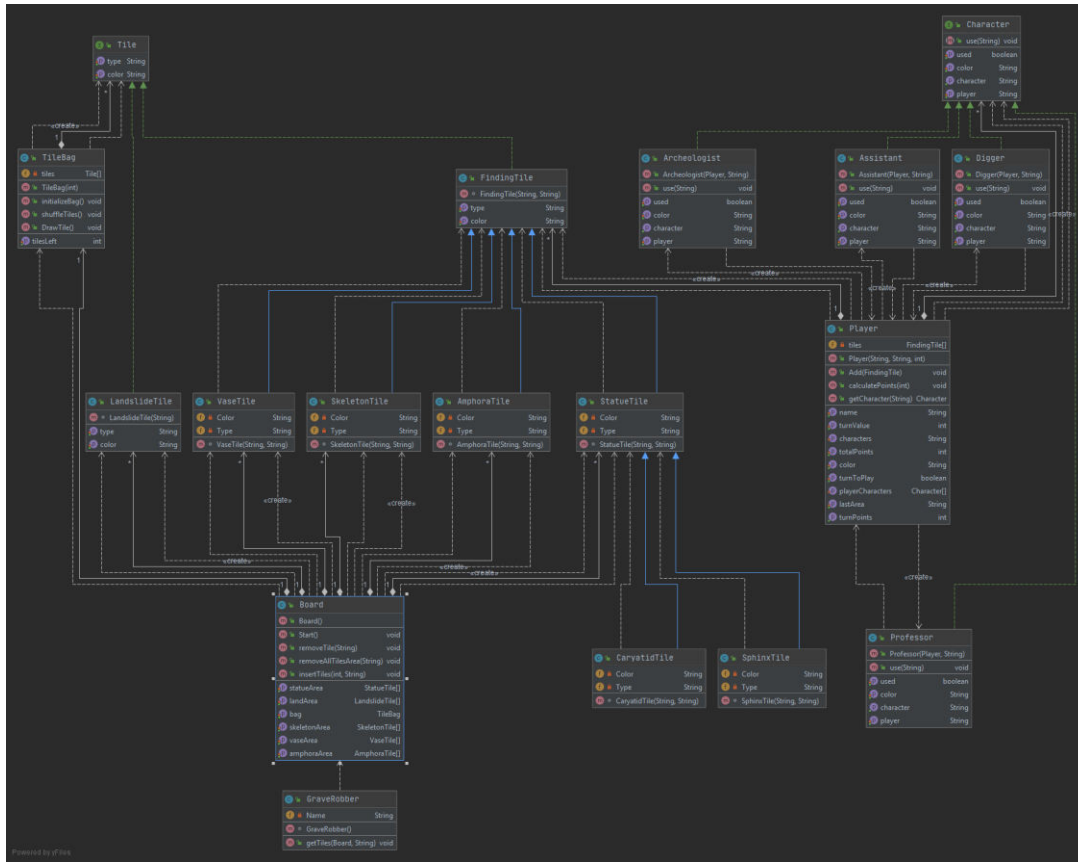












# ΤΕΛΟΣ