Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

OBJECT-ORIENTED PROGRAMMING HY-

PROJECT 2020-2021

Νίκολα Κοβάτσεβιτς

4285

12/01/2021

Περιεχόμενα

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 στο MODEL και το CONTROLLER, και στο τέλος στο VIEW.

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.

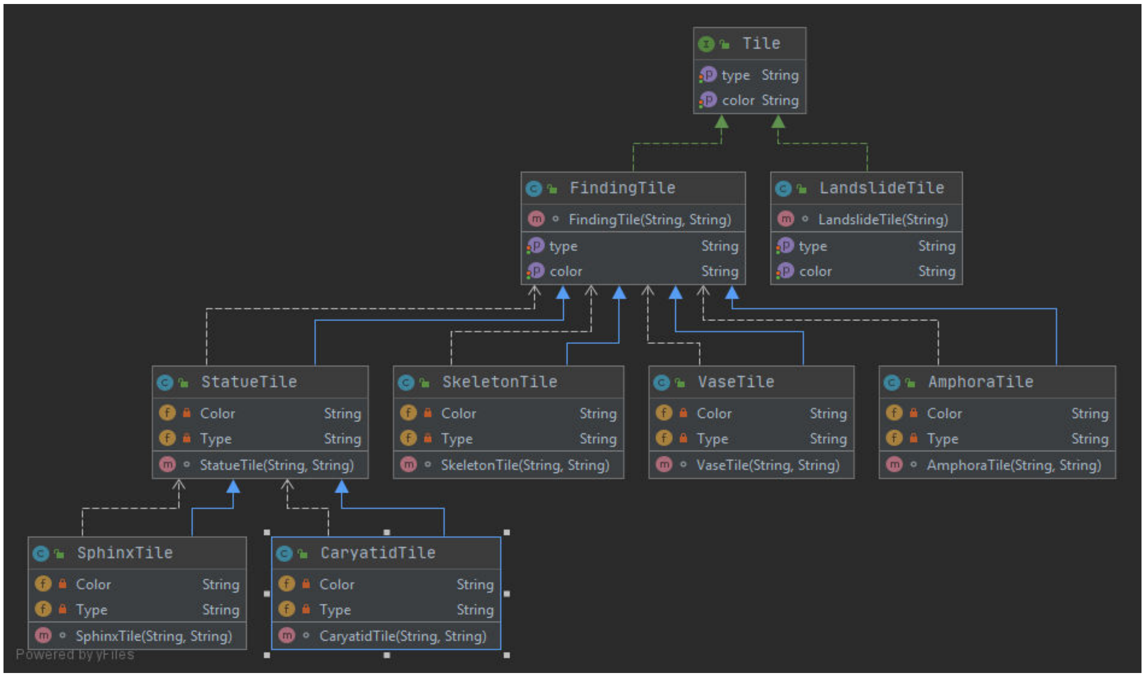
Classes ApmhoraTile,VaseTile,SkeletonTile,StatueTile

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 area1, int i); 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 player1; // 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

5. private Tile tile; // the tile it takes when used

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 area1, int i); Transformer(Mutative)

   Uses the card.

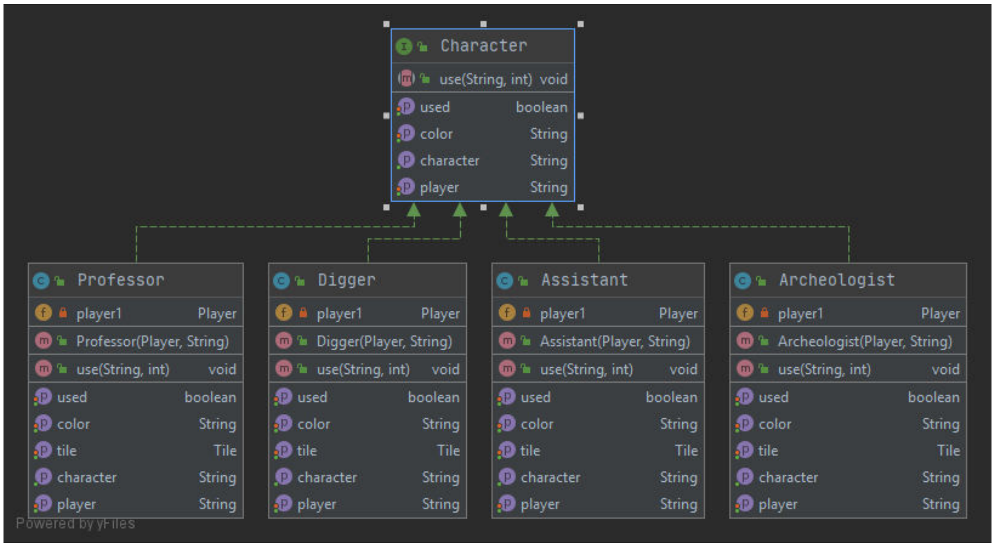
9. void setTile(Tile tile1); Transformer(Mutative)

   Sets the tile the player takes when using this card

10. Tile getTile(); Accessor(Selector)

    Returns the tile the player took using this card

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



Cl as s Pl a

yer

Η κλάση αυτή αναπαραστά τον παίκτη μέσα στο παιχνίδι.

Τα attributes

1. private String Name; // the name of the player
2. private Tile[] tiles = new Tile[111]; // the area where the player stores his tiles.
3. private Tile[] hand = new Tile[4]; // the player's hand
4. private String area; // the last area where the player took tiles from.
5. private int TotalPoints; // the total points of the player
6. private int turnPoints; // the points the player collects every turn.
7. private int turnValue; // the number of the current player's turn.
8. private String Color; // the color of the player.
9. private int TileNum; // thenumber of the tiles that the player holds
10. private int handSize; // the size of the player's hand
11. private int sphinxNum; // the number of Sphinx Tiles the player holds
12. private int CaryatidNum; // the number of Caryatid Tiles the player holds
13. private int timeToPlay; // value equals 1 if it's the player's turn to play 0

    // otherwise
14. private int isWinner; // value equals 1 if the player has won, 0 otherwise
15. private Character[] playerCharacters; // the character cards of the player

Οι μεθόδοι

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(Tile tile); Transformer(Mutative)

   Adds a tile to the player's tiles.
4. public void AddToHand(Tile tile); Transformer(Mutative)

Adds a tile to the player's hand.

5. public void setHandSize(int handSize); Transformer(Mutative)

   Changes the size of the player's hand

6. public void setTileNumber(int num); Transformer(Mutative)

   Sets the number of tiles the player holds in his area

7. public int getTileNumber(); Accessor(Selector)

   Returns the number of tiles the player holds in his area

8. public void setLastArea(String area); Transformer(Mutative)

   Sets the last area from which the player took a tile.

9. public String getLastArea(); Accessor(Selector)

   Returns the name of the last area from which the player took a tile.

10. public void calculatePoints(int points); Transformer(Mutative)

    Calculates the final points the player got in this game.

11. public int setTurnPoints(int turnPoints); Transformer(Mutative)

    Sets the points the player got.

12. public void findTurnPoints(); Transformer(Mutative)

    Calculates the points the player got.

13. public int getTurnPoints(); Accessor(Selector)

    Returns the points the player got.

14. public void setTotalPoints(int totalPoints); Transformer(Mutative)

    Sets the total points the player got in this game.

15. public int getTotalPoints(); Accessor(Selector)

    Returns the total points of the player.

16. public void setTurnValue(int turnValue); Transformer(Mutative)

    Sets the turn number that this player plays.

17. public int getTunrValue(); Accessor(Selector)

    Returns the turn number.

18. public void setColor(String color); Transformer(Mutative)

    Sets the color of the player.

19. public String getColor(); Accessor(Selector)

    Returns the color of the player.

20. public void setPlayerCharacters(Character[] playerCharacters);

    Transformer(Mutative)

    Sets the cards of the player.

21. public void setCharacters(string Color); Transformer(Mutative)

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

22. public Character getCharacter(String CharName); Accessor(Selector)

    Returns a certain player's card.

23. public void setSphinxNum(int sphinxNum); Transformer(Mutative)

    Sets the number of sphinx tiles the player holds.

24. public int getSphinxNum(); Accessor(Selector)

    Returns the number of sphinx tiles the player holds.

25. public void setCaryatidNum(int caryatidNum); Transformer(Mutative)

    Sets the number of caryatid tiles the player holds.

26. public int getCaryatidNum(); Accessor(Selector)

    Returns the number of caryatid tiles the player holds.

27. public Tile[] getHand(); Accessor(Selector)

    Returns the hand of the player.

28. public void setHandEmpty(); Transformer(Mutative)

    Empties the hand of the player.

29. public Tile[] getTiles(); Accessor(Selector)

    Returns the tiles that the player holds.

30. public void setTimeToPlay(int timeTopPlay); Transformer(Mutative)

    Sets the value of timeToPlay to 1 if it's the player's turn to play, 0 otherwise.

31. public int getTimeToPlay(); Accessor(Selector)

    Returns the timeToPlay value.

32. public void setIsWinner(int isWinner); Transformer(Mutative)

    Sets the value of isWinner to 1 if the player has won, 0 otherwise.

33. public int getIsWinner(); Accessor(Selector)

    Returns the value of isWinner.

Class GraveRobber

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

Τα attributes

1. private int totalPoints; // the final points of the GraveRobber
2. private int caryatidNum; // the number of caryatid tiles the GraveRobber holds.
3. private int sphynxTile; // the number of sphinx tiles the GraveRobber holds.
4. private Tile[] tiles; // the tiles the GraveRobber holds.
5. private int j; // used to access the tiles of the GraveRobber and insert new tiles
   //to him.

Οι μεθόδοι

1. public void getTiles(Board board); Transformer(Mutative)

   Removes all the tiles from the board except for the Landslide Tiles and adds them to his 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.

4. public void areaTotalPoints(); Transformer(Mutative)

   Calculates the points of the GraveRobber.

5. public void setCaryatidNum(int caryatidNum); Transformer(Mutative)

   Sets the number of caryatid tiles the GraveRobber holds.

6. public int getCaryatidNum(); Accessor(Selector)

   Returns the number of caryatid tiles the GraveRobber holds.

7. public void setSphinxNum(int sphinxNum); Transformer(Mutative)

   Sets the number of sphinx tiles the GraveRobber holds

8. public int getSphinxNum(); Accessor(Selector)

   Returns the number of sphinx tiles the GraveRobber holds.

Class TileBag

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

Τα 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.
3. private int players; // the number of the players.
4. private List<Tile> tileList; // A list of the tiles in the bag, used for shuffling the //bag.

Οι μέθοδοι

1. public void initializeBag(); Transformer(Mutative)

   Initializes the bag filling it with tiles.

2. public void shuffleTiles(); Transformer(Mutative)

   Shuffles the tiles inside the bag.

3. public void DrawTile(); 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.

6. public void delLandslide(int num); Transformer(Mutative)

   Removes a number of landslide tiles from the bag.

7. public Tile getDrawnTile(); Accessor(Selector)

   Returns the tile that is drawn from the bag.

8. public Tile[] getTiles(); Accessor(Selector)

   Returns the tiles that are in the bag.

Class Board

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

Τα attributes

1. private LansdlideTile[] 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.

7. private int lands; // the landslide tiles that get drawn out of the bag.

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, int pos); Transformer(Mutative)

    Removes a tile from a specific area.

15. public void insertTiles(String type, String color); Transformer(Mutative)

    Inserts a number of tiles in a given area.

16. public void setLands(int lands); Transformer(Mutative)

    Sets the number of the landslide tiles that get drawn out of the bag.

17. public int getLands(); Accessor(Selector)

    Returns the number of the landslide tiles that get draw out of the bag.

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 στιγμιοτύπων 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 Player[] players; // the players of the game.
4. private boolean gameOver; // if the game is over.
5. private boolean not_started; // if the game has not started yet.
6. private Player player1; // instance of player 1.
7. private Player player2; // instance of player 2.
8. private Player player3; // instance of player 3.

9. private Player player4; // instance of player4;

10. private int turn; // the current turn.

11. private GraveRobber r; // A GraveRobber instance.

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 turn numbers and assigns each of them to the players.

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 void drawTiles(Player player, TileBag bag); Transformer(Mutative)

    Draws 4 tiles from the bag for the player.

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

    Plays the player's turn.

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

    Sets a player as the winner of the game.

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

    Calculates the points to declare the winner.

17. Public Board getB(); Accessor(Selector)

    Returns the board of the game.

18. Public Player getPlayers(int turn); Accessor(Selector)

    Returns the player that has turn value equal with the parameter.

19. Public void setR(); Transformer(Mutative)

    Creates an instance of a GraveRobber.

20. Public GraveRobber getR(); Accessor(Selector)

    Returns an instance of a GraveRobber.

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Αυτό το πακέτο αποτελείται από τρις κλάσεις. Μια για το μενού των παικτών, μια για το μενού των attributes των παικτών, και μια για το παιχνίδι.

Class PlayerMenu

Η συγκεκριμένη κλάση αναπαριστά το μενού επιλογής του αριθμού των παικτών. Αυτή η κλάση επικοινωνεί άμεσα με την κλάση GrphicUI για να παρέχει τα αποτελέσματα της επιλογής του παίκτη/των παικτών και να μεταφερθούν μέσω αυτής της κλάσης στο Controller.

Τα attributes της κλάσης

1. private int players; // The number of players
2. private JDialog PlayerSelection; // The place where the contents will be visible,
   //it's attached to the frame.
3. Private JButton Player1; // This button will be pushed if the player wishes to
   //play solo.
4. Private JButton Player4; // This button will be pushed if the player wishes to
   //play multiplayer.

Οι μέθοδοι

1. public void setPlayers(int value); Transformer(Mutative)

   Sets the number of the players.

2. Public int getPlayers(); Accessor(Selector)

   Returns the number of the players.

3. Public void initializeButtons(); Transformer(Mutative)

   Initilizes the buttons of the menu.

4. Public void addButtons(); Transformer(Mutative)

   Adds the buttons to the frame.

5. Public void setActionListeners(); Transformer(Mutative)

   Sets the actionlisteners of the buttons so something will happen if they are pressed.

Class PlayerInfo

Αυτή η κλάση, καλείται αφού δοθεί ο αριθμός των παικτών μέσω του PlayerMenu, και χρησιμοποιείται για να δώσουμε στον κάθε παίκτη συγκεκριμένα στοιχεία, κυρίως όνομα και χρώμα. Η κλάση PlayerInfo όπως και η PlayerMenu επικοινωνεί άμεσα με την κλάση GraphicUI ώστε να μπορεί να παρέχει τα αποτελέσματα της στο Controller.

Τα attributes της κλάσης

1. private String Color; // The color of the player.
2. Private JDialog PlayerInfo; // The place where the contents of the frame will be //visible.
3. Private String Name; // The name of the player.(Never got used since we get the //player's name in the GraphicUi class)
4. Private JButton Black, Red, Blue, Green; // The buttons that will be shown to //the player

Οι μέθοδοι

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

   Sets the color of a player;

2. public String getColor(); Accessor(Selector)

   Returns the color of a player.

3. public void setName(String name); Transformer(Mutative)

   Sets the name of a player. (Never got used since we get the player's name in the GraphicUi class)

4. public String getName(); Accessor(Selector)

   Returns the name of a player. (Never got used since we get the player's name in the GraphicUi class)

5. public void initButtons(int num, String Last1, String Last2, String Last3); Transformer(Mutative)

   Initializes the buttons depending on which player is selecting.

6. Public void addButtons(int num, String Last1, String Last2, String Last3); Transformer(Mutative)

   Adds the buttons to the frame depending on which player is selecting.

7. Public void setActionListeners(int num, String Last1, String Last2, String Last3); Transformer(Mutative)

   Adds actionListeners to the buttons depending on which player is selecting.

Class GraphicUI

Η κλάση GraphicUI είναι η κεντρική κλάση του πακέτου View. Η κλάση αυτή επικοινωνεί άμεσα με όλες τις υπόλοιπες κλάσεις του πακέτου, καθώς και με την κλάση Controller. Η κλάση αυτή είναι υπεύθυνη για τον σχεδιασμό των γραφικών που βλέπει ο χρήστης, των ήχων που ακούει καθώς και για την αλληλεπίδραση μεταξύ αυτού και του προγράμματος.

Στην συγκεκριμένη περίπτωση η κλάση αυτή είναι υπεύθυνη για την δημιουργία των γραφικών του αρχικού μενού, του ταμπλό του παιχνιδιού, των κουμπιών για τράβηγμα από την σακούλα, τέλος του γύρου, καινούριο παιχνίδι, σώσιμο του παιχνιδιού, συνέχεια του παιχνιδιού, και έξοδος από το πρόγραμμα, καθώς και για τα πλακίδια, τις κάρτες τις κινήσεις, αλλά και την περιοχή όπου ο παίκτης κρατάει τα πλακίδια που πήρε, όπως και γα την αναγραφή του νικητή.

Τα attributes της κλάσης

1. private Controller game; // instance of the controller of the game.
2. Private JButton PlayButton, Start; // You start a new game by pressing these //buttons
3. private JButton ExitButton, Exit; // You exit the game by pressing these //buttons
4. private JButton continueButton, Cont; //You can continue a saved game by / //pressing this button(not implemented //due to time constraints)
5. private JButton SaveButton; // You can save your game by pressing this //button(not implemented due to time constraints)
6. private JButton[] Buttons; // The buttons of the tiles;
7. private JButton[] Cards; // The buttons for the cards;
8. private JLabel[] us; // Labels with gray color that get in front of used cards.
9. Private JButton bag; // The button from which the tiles are drawn(initially //showed the word "bag" but later changed it to "Draw" //for confusion solving purposes)
10. private JButton endTurn; // The button to end the turn;
11. private JLayeredPane Playing_field; // A layered pane that holds all the //contents of the frame.
12. Private JLabel Background; //The background image of the title screen.
13. Private JLabel table; // The background of the main screen.

➔ Σημείωση : (Στην εκφώνηση δεν εξηγούταν ακριβώς πώς θα έπρεπε να εμφανίζονται τα πλακίδια που παίρνει ο κάθε παίκτης και που, έτσι αποφάσισα να βάλω όλους τους διαφορετικούς τύπους πλακιδίων σαν εικόνες στο κάτω μέρος της οθόνης, με μετρητές στην πάνω δεξιά άκρη της κάθε εικόνας.)

➔ Έτσι κάθε JLabel που θα αναγραφεί από την αρχή μέχρι το τέλος των άγκιστρων στην συνέχεια θα αντιπροσωπεύει αυτές τις εικόνες και τους μετρητές.

{

Εικόνες:

14. private JLabel MosaicGreen, MosaicRed, MosaicYellow, StatCaryatid, StatSphinx, AmphoraBlue, AmphoraBrown, AmphoraRed, AmphoraGreen, AmphoraYellow, AmphoraPurple, UpperBody, LowerBody, UpperSmallBody, LowerSmallBody.

Μετρητές:

15. private JLabel MGT, MRT, MYT, CT, ST, ABT, ABrT, ART, AGT, AYT, APT, UBT, LBT, UsBT, LsBT

}

16. private URL imageURl; // Holds an imge.

17. Private ClassLoader cldr; // passes the URL of an image to imageURL.

18. Private final GraphicUI ui; // Instance of the current class, used to make things
//like JoptionPanes appear on the screen.

19. Private int size; // The size of the JButton array

➔ Σημείωση: (Τα παρακάτω integers αναπαριστούν συντεταγμένες για τα πλακίδια στον άξονα X και Y.)

20. private int MosaicX, MosaicY, StatueX, StatueY, AmphoraX, AmphoraY, SkeletonX, SkeletonY.

21. Private int tilesTaken; // The number of tiles the player took this turn.

22. Private JButton tilesInBoard; //This array holds the tiles that are currently on
//the board.

23. Private int in; // the number of the current player.

24. private int card; // Has the value of 1 if a card is being used, 0 otherwise.

25. Private String area; // The last area the player took a tile from(without using
//cards)

26. private String Character; // The last character the player used.

27. Private String[] areas; // An array that hold names of different areas.
    //Exclusively used by the professor card.

28. Private int S, M, A, Skel; //Counters that are used by the professor card to show
    //if the player has already taken a tile from a specific
    //area.

29. Private int done; //Takes the value of 0 if the player has taken all the tiles he
    //can by using a card, otherwise it takes the value of the tiles
    //he has left to take.

30. Private int Nturn; //Takes the value of 1 if the player has pressed the bag
    //button, 0 if he pressed the endTurn button.

31. Private int[] P1tiles; // holds the numbers of each type of tile the player 1 owns.

32. Private int[] P2tiles; // holds the numbers of each type of tile the player 2 owns.

33. Private int[] P3tiles; // holds the numbers of each type of tile the player 3 owns.

34. Private int[] P4tiles; // holds the numbers of each type of tile the player 4 owns.

35. Private Clip Mg; // Clip instance for the title soundtrack.

36. Private AudioStream bgm; // AudioStream instance for the soundtrack.

37. Private Clip ng; // Clip instance for the in-game soundtrack.

Οι μέθοδοι

1. public void InitComponents(); Transformer(Mutative)

   Initializes the components of the title screen.

2. Public void InitButtons();Transformer(Mutative)

   Initializes the buttons and components for the main screen.

3. Public class Playlistener;

   Actionlistener for the Start button and the PlayGame button. It makes these buttons create a new game upon clicking. It calls the PlayerMenu and PlayerInfo so it can notify the controller of the game with the results.

4. Public class EndTurnListener;

   Actionlistener for the endTurn button. It makes it so the turn ends uppon clicking the button, and passes on the information for the next player.

5. Public class BagListener;

Actionlistener for the bag button. It makes this button draw the next tile from the bag when pressed, and places the tiles in their respective areas.

6. Public class SaveListener;

   Not implemented

7. public class ContinueListener;

   Not implemented

8. public class CardListener;

   Actionlistener for the cards. It makes the card buttons notify the TileListener that a card has been used and used the card.

9. Public class TileListener;

   Actionlistener for the tiles. It makes the tile buttons disappear when pressed and get in the area of the player who pressed them. It also shows a warning message if the player tries to take more tiles than he should.

10. Public class ExitListener;

    Actionlistener for the Exit and ExitButton buttons. It makes the buttons exit the program when pressed.

➔ Σημείωση: (οι παρακάτω συναρτήσεις θέτουν και επιστρέφουν τις συντεταγμένες X και Y των πλακιδίων.)

11. private void setMosaicX(int mosaicX), setStatueX(int statueX), setAmphoraX(int amphoraX), setSkeletonX(int skeletonX), setLandslidX(int landslideX) setMosaicY(int mosaicY), setStatueY(int statueY), setAmphoraY(int amphoraY), setSkeletonY(int skeletonY), setLandslideY(int landslideY);

12. private int getMosaicX(), getStatueX(), getAmphoraX(), getSkeletonX(), getLandslideX(), get MosaicY(), getStatueY(), getAmphoraY(), getSkeletonY(), getLandslideY();

13. private void setTilesTaken(int tiles);

    Sets the number of the tiles that the player took this turn.

14. Private int getTilesTaken();

    Returns the tiles the player took this turn.

15. Public void changeText(JLabel label, String text);

    Changes the text of a given label.

16. Public void setArea(String ar);

    Sets the area to the last area the player took a tile from.

17. Public String getArea();

    Returns the last area the player took a tile from.

18. Public void setCharacter(String Character);

    Sets the last character the player used.

19. Public String getCharacter();

    Returns the last character the player used.

20. Public void setIn(int in);

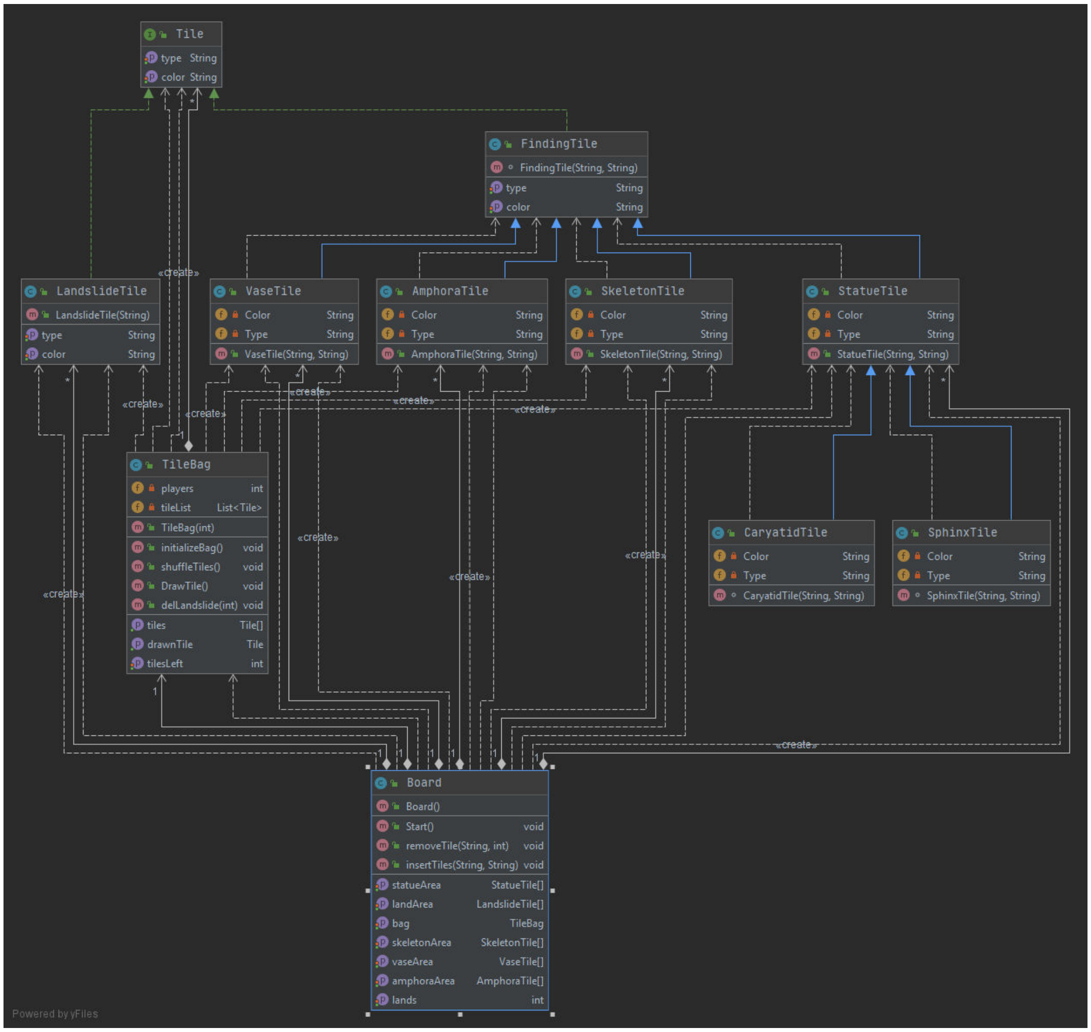
    Sets the number of the current player.

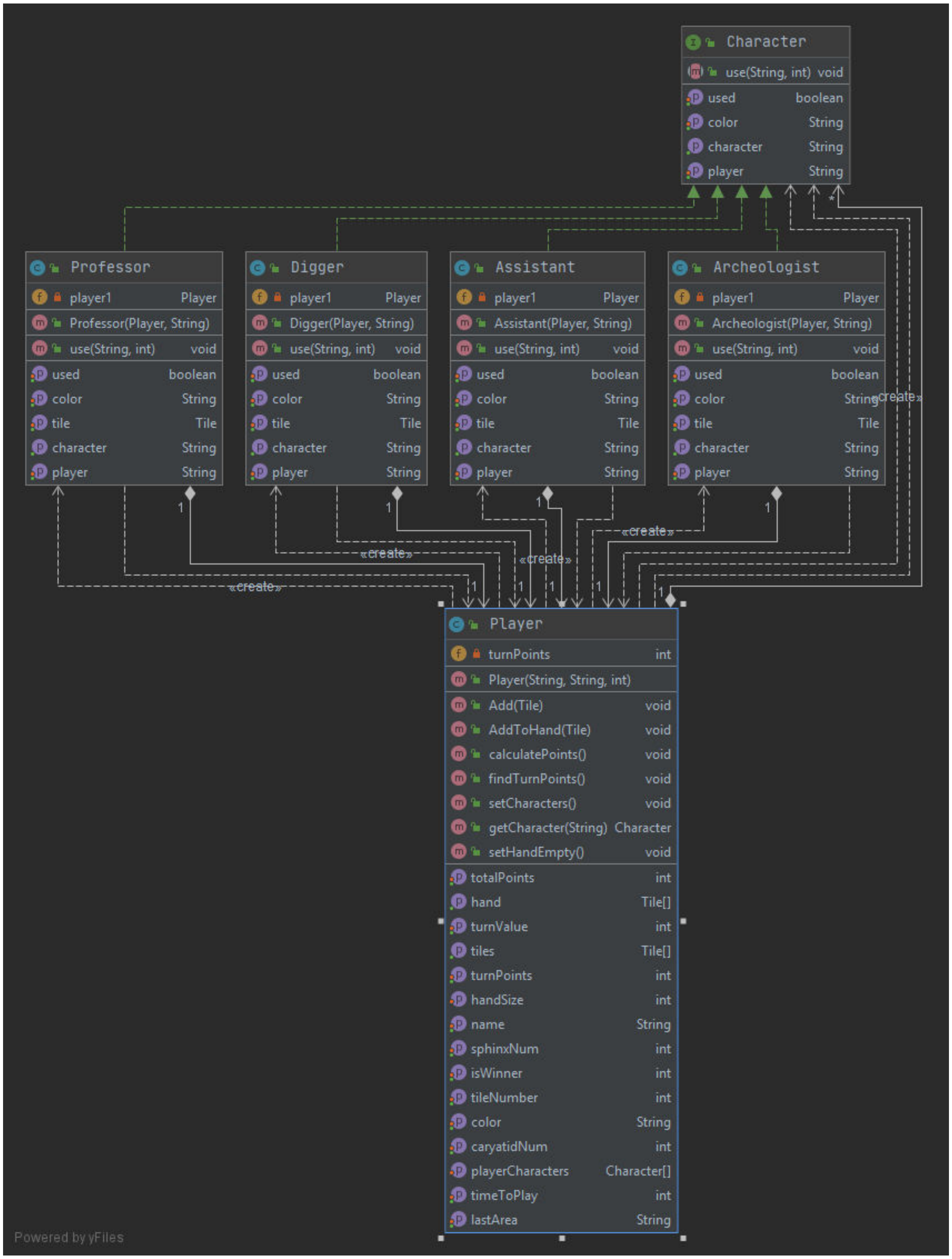
21. Public int getIn();

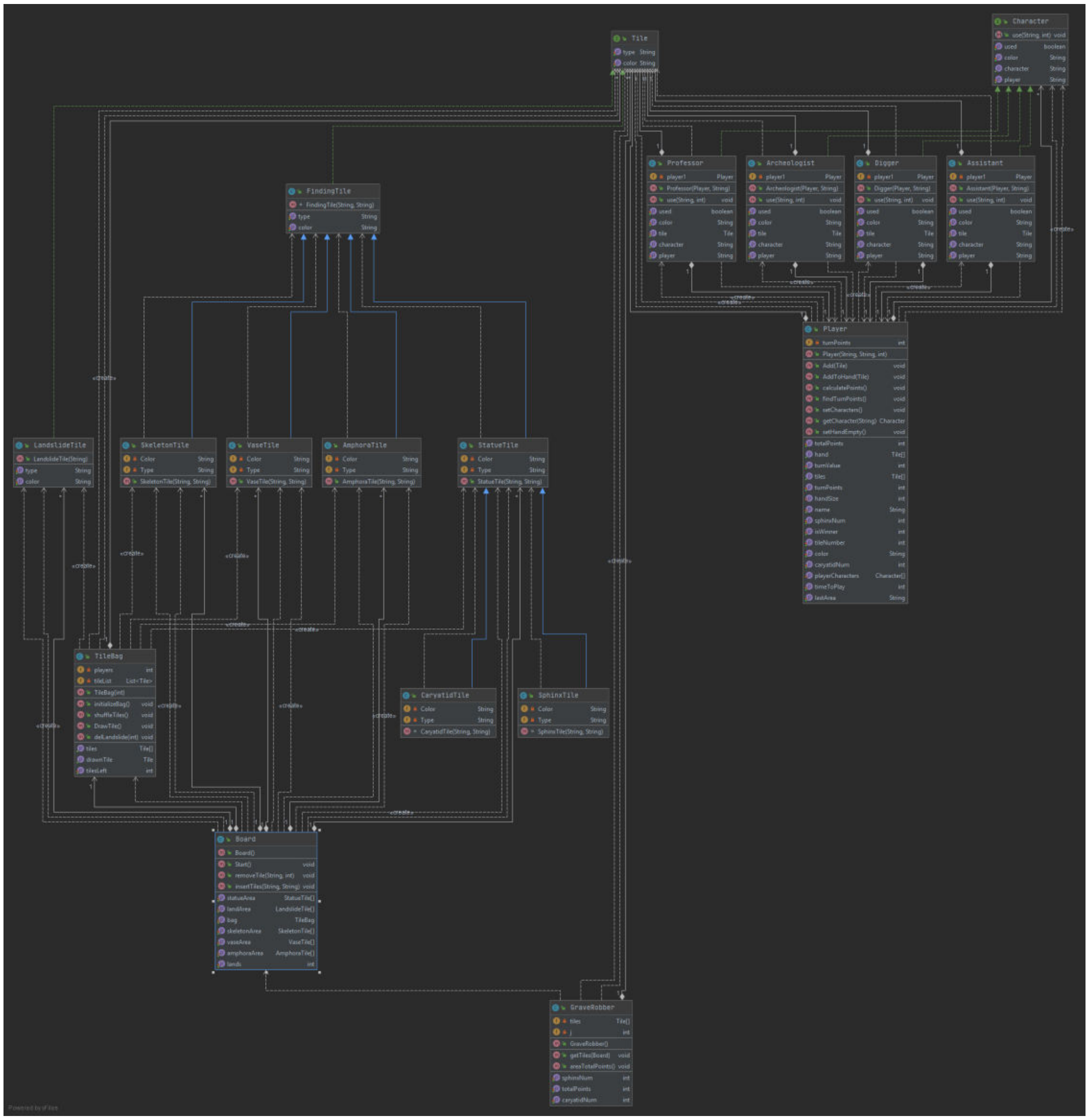
    Returns the number of the current player.

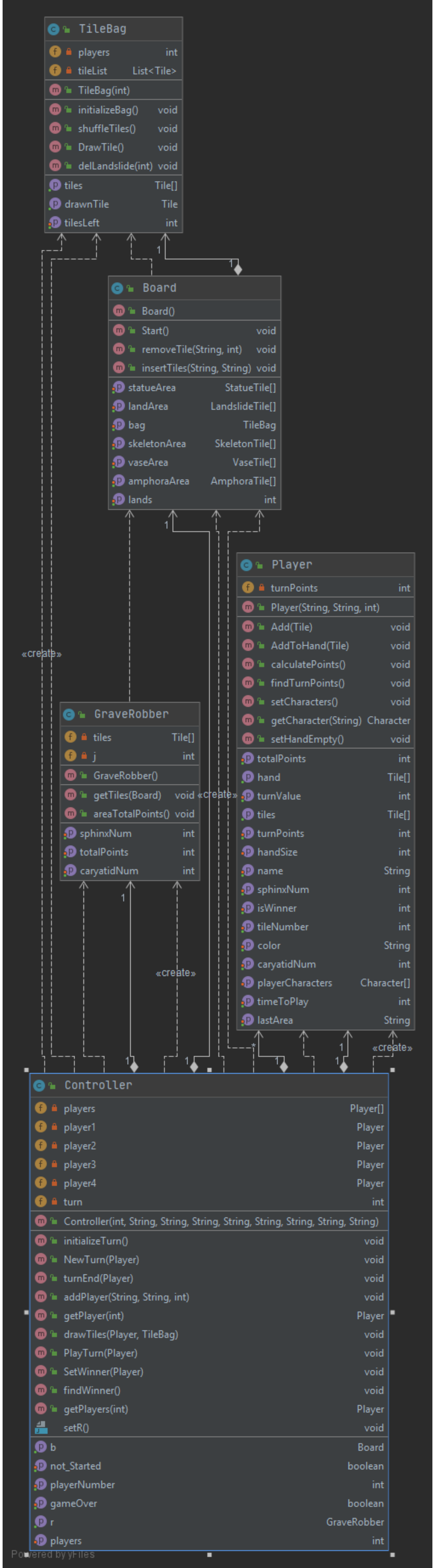
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

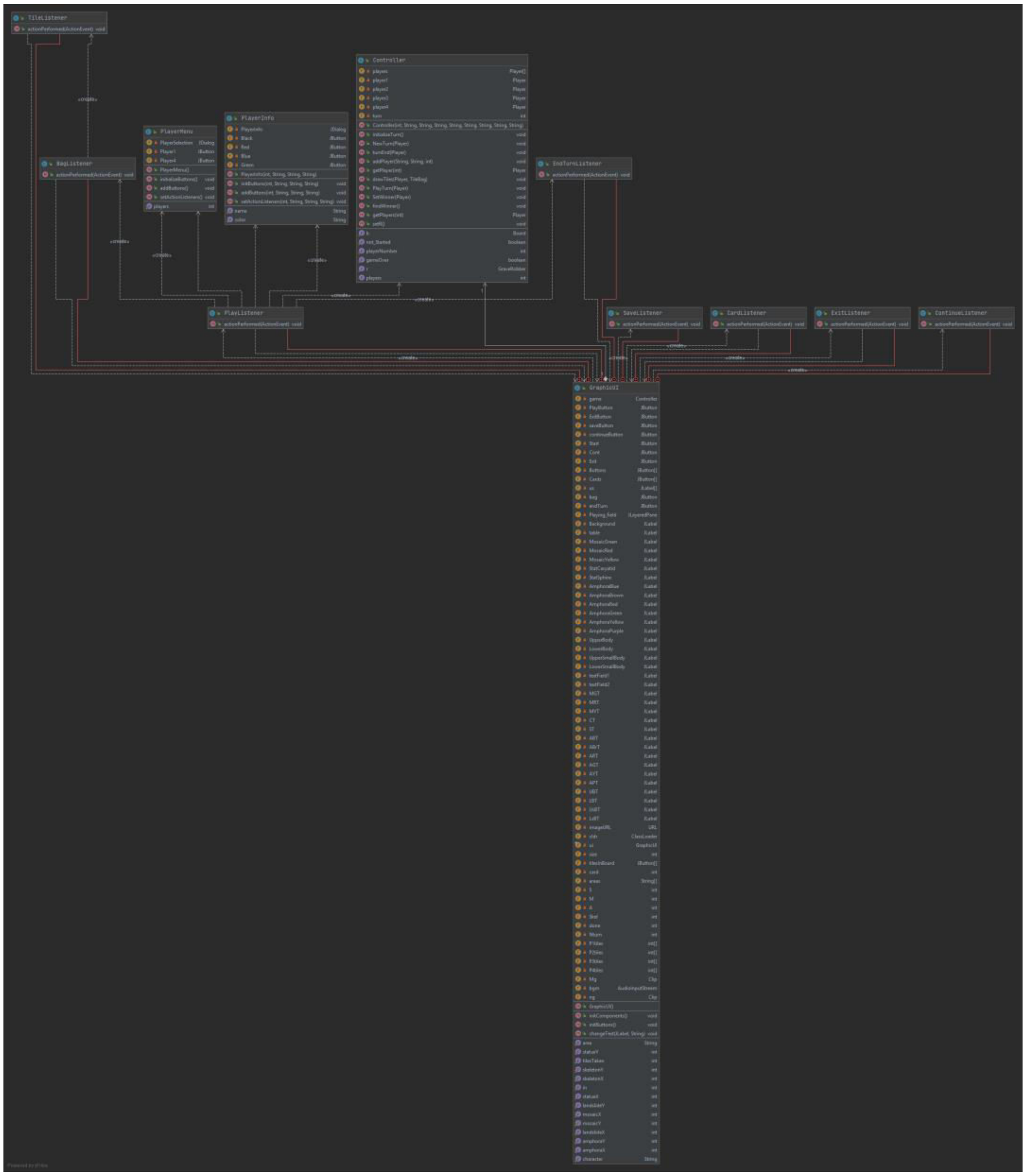
Στην συνέχεια θα συμπεριληφθούν τα διαγράμματα UML για την αλληλεπίδραση μεταξύ κλάσεων και πακέτων.

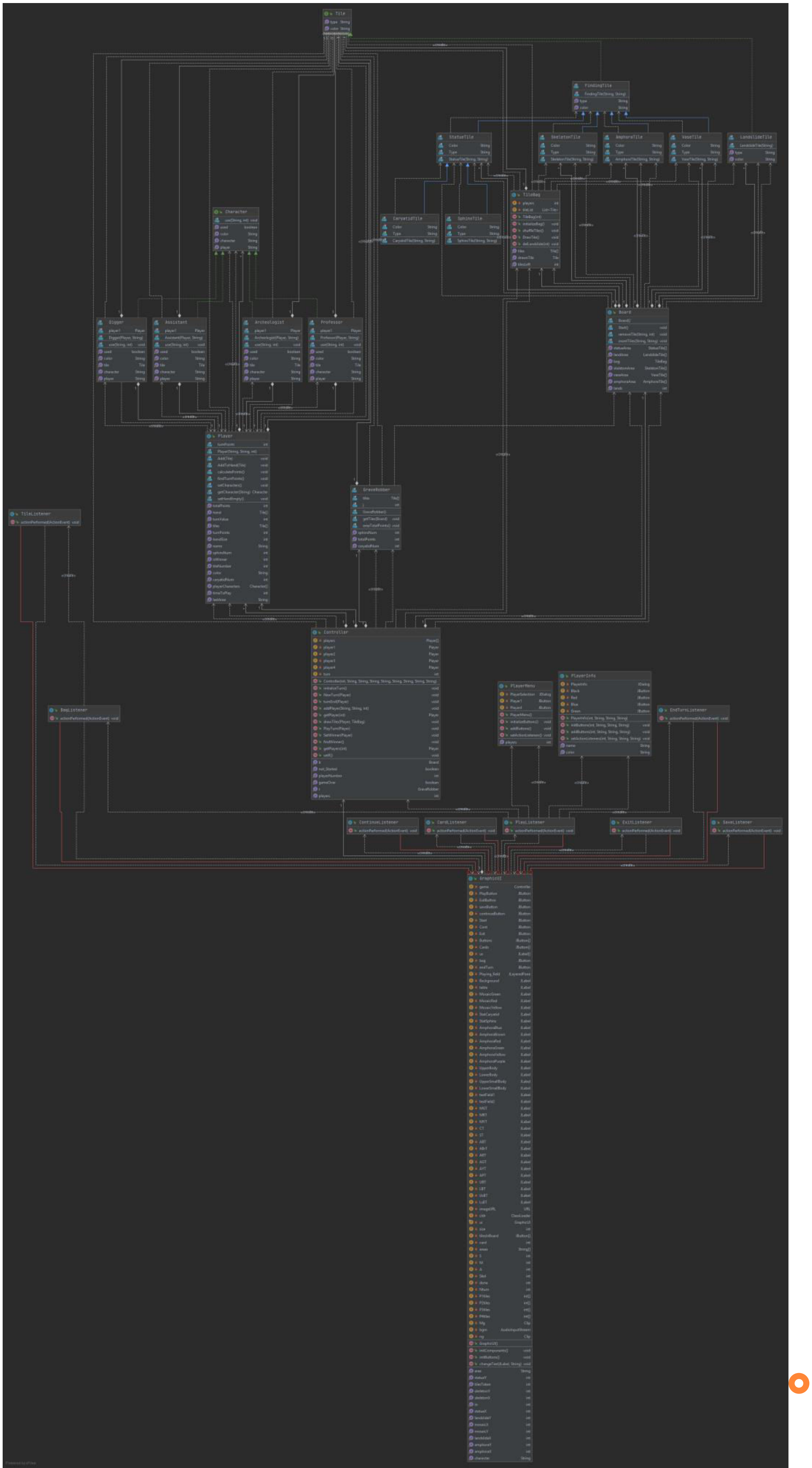

Tile
type String
color String
FindingTile
FindingTile(String, String)
type String
color String
LandslideTile
LandslideTile(String)
type String
color String
VaseTile
Color String
Type String
VaseTile(String, String)
AmphoraTile
Color String
Type String
AmphoraTile(String, String)
SkeletonTile
Color String
Type String
SkeletonTile(String, String)
StatueTile
Color String
Type String
StatueTile(String, String)
TileBag
players int
tileList List<Tile>
TileBag(int)
initializeBag() void
shuffleTiles() void
DrawTile() void
delLandslide(int) void
tiles Tile[]
drawnTile Tile
tilesLeft int
CaryatidTile
Color String
Type String
CaryatidTile(String, String)
SphinxTile
Color String
Type String
SphinxTile(String, String)
Board
Board()
Start() void
removeTile(String, int) void
insertTiles(String, String) void
statueArea StatueTile[]
landArea LandslideTile[]
bag TileBag
skeletonArea SkeletonTile[]
vaseArea VaseTile[]
amphoraArea AmphoraTile[]
lands int
«create»
*
1
Powered by yFiles


Character
use(String, int) void
used boolean
color String
character String
player String
Professor
player1 Player
Professor(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Digger
player1 Player
Digger(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Assistant
player1 Player
Assistant(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Archeologist
player1 Player
Archeologist(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
«create»
«create»
«create»
«create»
«create»
1
1
1
1
1
1
1
1
1
Player
turnPoints int
Player(String, String, int)
Add(Tile) void
AddToHand(Tile) void
calculatePoints() void
findTurnPoints() void
setCharacters() void
getCharacter(String) Character
setHandEmpty() void
totalPoints int
hand Tile[]
turnValue int
tiles Tile[]
turnPoints int
handSize int
name String
sphinxNum int
isWinner int
tileNumber int
color String
caryatidNum int
playerCharacters Character[]
timeToPlay int
lastArea String
Powered by yFiles


Tile
type String
color String
Character
use(String, int) void
used boolean
color String
character String
player String
Professor
player1 Player
Professor(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Archeologist
player1 Player
Archeologist(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Digger
player1 Player
Digger(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
Assistant
player1 Player
Assistant(Player, String)
use(String, int) void
used boolean
color String
tile Tile
character String
player String
FindingTile
FindingTile(String, String)
type String
color String
Player
turnPoints int
Player(String, String, int)
Add(Tile) void
AddToHand(Tile) void
calculatePoints() void
findTurnPoints() void
setCharacters() void
getCharacter(String) Character
setHandEmpty() void
totalPoints int
hand Tile[]
turnValue int
tiles Tile[]
turnPoints int
handSize int
name String
sphinxNum int
isWinner int
tileNumber int
color String
caryatidNum int
playerCharacters Character[]
timeToPlay int
lastArea String
LandslideTile
LandslideTile(String)
type String
color String
SkeletonTile
Color String
Type String
SkeletonTile(String, String)
VaseTile
Color String
Type String
VaseTile(String, String)
AmphoraTile
Color String
Type String
AmphoraTile(String, String)
StatueTile
Color String
Type String
StatueTile(String, String)
TileBag
players int
tileList List<Tile>
TileBag(int)
initializeBag() void
shuffleTiles() void
DrawTile() void
delLandslide(int) void
tiles Tile[]
drawnTile Tile
tilesLeft int
CaryatidTile
Color String
Type String
CaryatidTile(String, String)
SphinxTile
Color String
Type String
SphinxTile(String, String)
Board
Board()
Start() void
removeTile(String, int) void
insertTiles(String, String) void
statueArea StatueTile[]
landArea LandslideTile[]
bag TileBag
skeletonArea SkeletonTile[]
vaseArea VaseTile[]
amphoraArea AmphoraTile[]
lands int
GraveRobber
tiles Tile[]
j int
GraveRobber()
getTiles(Board) void
areaTotalPoints() void
sphinxNum int
totalPoints int
caryatidNum int
«create»
1
Powered by yFiles


TileBag
players int
tileList List<Tile>
TileBag(int)
initializeBag() void
shuffleTiles() void
DrawTile() void
delLandslide(int) void
tiles Tile[]
drawnTile Tile
tilesLeft int
1
1
Board
Board()
Start() void
removeTile(String, int) void
insertTiles(String, String) void
statueArea StatueTile[]
landArea LandslideTile[]
bag TileBag
skeletonArea SkeletonTile[]
vaseArea VaseTile[]
amphoraArea AmphoraTile[]
lands int
1
Player
turnPoints int
Player(String, String, int)
Add(Tile) void
AddToHand(Tile) void
calculatePoints() void
findTurnPoints() void
setCharacters() void
getCharacter(String) Character
setHandEmpty() void
totalPoints int
hand Tile[]
turnValue int
tiles Tile[]
turnPoints int
handSize int
name String
sphinxNum int
isWinner int
tileNumber int
color String
caryatidNum int
playerCharacters Character[]
timeToPlay int
lastArea String
«create»
GraveRobber
tiles Tile[]
j int
GraveRobber()
getTiles(Board) void
areaTotalPoints() void
sphinxNum int
totalPoints int
caryatidNum int
«create»
1
«create»
1
1
*
1
1
«create»
Controller
players Player[]
player1 Player
player2 Player
player3 Player
player4 Player
turn int
Controller(int, String, String, String, String, String, String, String, String)
initializeTurn() void
NewTurn(Player) void
turnEnd(Player) void
addPlayer(String, String, int) void
getPlayer(int) Player
drawTiles(Player, TileBag) void
PlayTurn(Player) void
SetWinner(Player) void
findWinner() void
getPlayers(int) Player
setR() void
b Board
not_Started boolean
playerNumber int
gameOver boolean
r GraveRobber
players int
Powered by yFiles


TileListener
actionPerformed(ActionEvent) void
BagListener
actionPerformed(ActionEvent) void
PlayerMenu
PlayerInfo
Controller
EndTurnListener
actionPerformed(ActionEvent) void
PlayListener
actionPerformed(ActionEvent) void
SaveListener
CardListener
ExitListener
ContinueListener
«create»
GraphicUI


Tile
FindingTile
StatueTile
SkeletonTile
AmphoraTile
VaseTile
LandslideTile
CaryatidTile
SphinxTile
TileBag
Character
Digger
Assistant
Archeologist
Professor
Board
Player
GraveRobber
TileListener
Controller
PlayerMenu
PlayerInfo
EndTurnListener
BagListener
ContinueListener
CardListener
PlayListener
ExitListener
SaveListener
GraphicUI

ΤΕΛΟΣ