



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

Εισαγωγή:

Σε αυτο το project δημιουργουμε το επιτραπεζιο αμφιπολη σε ηλεκτρονικη μορφη. Ειμαστε ακομη στην πρωτη φαση της υλοποιησης του αλλα με το περας και της δευτερης φασης θα εχουμε μια ολοκληρωμενη εικονα του project με πληρες λειτουργικοτητα και GUI.

Περιεχόμενα:

1. Εισαγωγή	2
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model	2.....
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller	11.....
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View	13.....
5. Η Αλληλεπίδραση μεταξύ των κλάσεων - Διαγράμματα UML	14.....
6. Λειτουργικότητα (Β Φάση)	17
7. Συμπεράσματα	17

- Εισαγωγή

Το project αυτο βασίζεται στο σχεδιαστικό μοντελο MVC (model - view - controler), το οποιο συνοπτικα χωριζει το project σε 3 διαφορετικες λειτουργιες: α) Model: τα δεδομενα (τροπος παιχνιδιου,παιχτες κτλ), β) View: η γραφικη διεπαφη ή αλλιως GUI, γ) Controler: ο συνδετικος κρικος των δυο παραπανω. Οι ενοτητες της ανοφορας αυτης περιλαμβανουν την αναλυση του MVC που χρησιμοποιηθηκε, με σχολια πανω στις λειτουργιες καθε μεθοδου αλλα και UML διαγραμματα για την γραφικη απεικονιση του project.

- Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακετο model το χωρισα σε 4 "βασικα" πακετα: α) Tile, β) Player, γ) Characters, δ) Board. Καθε πακετο εχει μια συγκεκριμενη δουλεια για την σωστη υλοποιηση του πακετου model.

Πακετο Tile:

Σε αυτο το πακετο περιεχονται ολες οι μεθοδοι - κλασεις που ειναι υπευθυνες για την αρχικοποιηση των Tiles του παιχνιδιου.

Η abstract κλαση Tile: ρολος αυτης της κλασης ειναι η δημιουργια πλακιδιων απο διαφορους τυπους. Καθε ενας απο αυτους τους τυπους θα κανει extend την κλαση Tile (δείτε σχημα 1).

Οι μεθοδοι της Tile ειναι:

1. public Tile(String type); Αυτος ειναι ο constructor της κλασης. Δουλεια του ειναι η δημιουργια ενος καινουργιου Tile με "type" τυπο.
2. public String getType(); Αυτος ειναι ενας Accessor ο οποιος επιστρεφει τον τυπο που αντιστοιχει στο Tile.
3. public void setType(String Type); Αυτος ειναι ενας Transformer ο οποιος αρχικοποιει τον τυπο του Tile σε Type;

Τα attributes της Tile είναι:

1. private String type; Ο τυπος που αντιστοιχει σε καθε Tile.

Η κλαση Bag: Ο ρολος της κλασης αυτης ειναι η δημιουργια μιας καινουργιας τσαντας με ολα τα Tiles αρχικοποιημενα για το παιχνιδι, οπως και η παροχη μεθοδων για την διαχειριση της.

Οι μεθοδοι της Bag ειναι:

1. public Bag(); Αυτος ειναι ο constructor της κλασης. Δουλεια του ειναι η δημιουργια μιας καινουργιας Bag και η αρχικοποιηση της λιστας TotalTiles.
2. public void initializeTiles(); Αυτος ειναι ενας Transformer ο οποιος αρχικοποιει ολα τα Tiles του παιχνιδιου και τα προσθετει στην TotalTiles λιστα.
3. public void AddTile(Tile t); Αυτος ειναι ενας Observer ο οποιος προσθετει ενα Tile t στην λιστα TotalTiles.
4. public Tile RemoveFirstTile(); Αυτος ειναι ενας Accessor ο οποιος αφαιρει το πρωτο Tile απο την λιστα TotalTiles.

Τα attributes της Bag ειναι:

1. private List <Tile> TotalTiles; Η λιστα με ολα τα αρχικοποιημενα Tiles του παιχνιδιου.

Η abstract κλαση FindingTile: Η κλαση αυτη υπαρχει για την ομαδοποιηση των Tiles παρομοιου τυπου. Για παραδειγμα, extend αυτης της κλασης ειναι τα Amphora, Skeleton, Mosaic και Statue Tiles.

Οι μεθοδοι της FindingTile ειναι:

1. public FindingTile(String type); Αυτος ειναι ο constructor της κλασης. Δουλεια του ειναι η δημιουργια ενος καινουργιου FindingTile με “type” τυπο.

Η κλαση LandslideTile: Η κλαση αυτη ειναι υπευθυνη για την δημιουργια ενος LandslideTile.

Οι μεθοδοι της LandslideTile ειναι:

1. public LandslideTile(String type); Αυτος ειναι ο constructor της κλασης, ο οποίος περνει σαν ορισμα τον τυπο “type” του Tile και δημιουργει ενα LandslideTile.

Η κλαση AmphoraTile: Η κλαση αυτη δημιουργει ενα καινουργιο AmphoraTile.

Οι μεθοδοι της AmphoraTile ειναι:

1. public AmphoraTile(String type, String color); Αυτος ειναι ο constructor της AmphoraTile. Δουλεια του ειναι η δημιουργια ενος καινουργιου AmphoraTile με “type” τυπο και “color” χρωμα.

Τα attributes της AmphoraTile ειναι:

1. private String color; Το χρωμα που αντιστοιχει στο AmphoraTile.

Η κλαση SkeletonTile: Η κλαση αυτη δημιουργει ενα καινουργιο SkeletonTile.

Οι μεθοδοι της SkeletonTile ειναι:

1. public SkeletonTile (String type, int category, int part); Αυτος ειναι ο constructor της SkeletonTile. Δουλεια του ειναι η δημιουργια ενος καινουργιου SkeletonTile με “type” τυπο , “part” και “category”.

Τα attributes της SkeletonTile ειναι:

1. private int part; Το μερος σωματος (πανω / κατω).

2. private int category; Κατηγορια ανθρωπου (ενηλικας / παιδι).

Η κλαση MosaicTile: Η κλαση αυτη δημιουργει ενα καινουργιο MosaicTile.

Οι μεθοδοι της MosaicTile ειναι:

1. public MosaicTile(String type, String color); Αυτος ειναι ο constructor της MosaicTile. Δουλεια του ειναι η δημιουργια ενος καινουργιου MosaicTile με “type” τυπο και “color” χρωμα.

Τα attributes της MosaicTile ειναι:

1. private String color; Το χρωμα που αντιστοιχει στο MosaicTile.

Η abstract κλάση StatueTile: Η κλάση αυτή υπάρχει για την ομαδοποίηση των Tiles παρομοίου τύπου. Για παράδειγμα, extend αυτής της κλάσης είναι τα Caryatid και Sphinx Tiles.

Οι μέθοδοι της StatueTile είναι:

1. public StatueTile(String type); Αυτός είναι ο constructor της StatueTile. Δουλειά του είναι η δημιουργία ενός καινούργιου StatueTile με “type” τύπο.

Η κλάση CaryatidTile: Η κλάση αυτή δημιουργεί ένα καινούργιο CaryatidTile.

Οι μέθοδοι της CaryatidTile είναι:

1. public CaryatidTile(String type); Αυτός είναι ο constructor της CaryatidTile. Δουλειά του είναι η δημιουργία ενός καινούργιου CaryatidTile με “type” τύπο.

Η κλάση SphinxTile: Η κλάση αυτή δημιουργεί ένα καινούργιο SphinxTile.

Οι μέθοδοι της SphinxTile είναι:

1. public SphinxTile(String type); Αυτός είναι ο constructor της SphinxTile. Δουλειά του είναι η δημιουργία ενός καινούργιου SphinxTile με “type” τύπο.

Πακέτο Player:

Σε αυτό το πακέτο περιέχονται όλες οι μέθοδοι - κλάσεις που είναι υπεύθυνες για την αρχικοποίηση των Players του παιχνιδιού.

Η κλάση Player: Αυτή η κλάση είναι υπεύθυνη για την αποθήκευση και διαχείριση των παικτών του παιχνιδιού.

Οι μέθοδοι της Player είναι:

1. public Player(String name, int id); Αυτός είναι ο constructor της Player. Δουλειά του είναι η δημιουργία ενός καινούργιου Player με “name” όνομα και “id” ταυτότητα / αναγνωριστικό. Επίσης αρχικοποιεί την PlayerTiles και CharacterTiles List όπως και τα played, finished και action του παίκτη.

2. public void changePlayed(boolean b); Αυτός είναι ένας Transformer ο οποίος αλλάζει την τιμή played του παίκτη σε “b”, όπου “b” είναι μια boolean μεταβλητή.

3. `public void changeFinished(boolean b);` Αυτος ειναι ενας Transformer ο οποίος αλλαζει την τιμη `finished` του παιχτη σε `"b"`.

4. `public String getName();` Αυτος ειναι ενας Accessor, ο οποίος επιστρεφει το ονομα του παιχτη.

5. `public void setName(String name);` Αυτος ειναι ενας Transformer, ο οποίος αλλαζει το ονομα του παιχτη σε `"name"`.

6. `public void AddToStash(Tile t);` Αυτος ειναι ενας Transformer, ο οποίος προσθετει το Tile `"t"` στο `stash` του παιχτη.

7. `public void AddCharacter(Character c);` Αυτος ειναι ενας Transformer, ο οποίος προσθετει τον Character `"c"` στο `stash` του παιχτη.

8. `public void RemoverCharacter(Character c);` Αυτος ειναι ενας Transformer, ο οποίος αφαιρει τον Character `"c"` στο `stash` του παιχτη.

9. `public int CountPoints();` Αυτος ειναι ενας Accessor, ο οποίος μετραει ολους τους ποντους του παιχτη.

10. `public void setAction(int action);` Αυτος ειναι ενας Transformer, ο οποίος αλλαζει το `action` του παιχτη σε `"action"`.

Τα attributes της Player ειναι:

1. `List <Tile> PlayerTiles;` Η λιστα με ολα τα Tile που εχει συγκεντρωσει ο παιχτης
2. `List <Character> PlayerCharacters;` Η λιστα με ολους τους Characters που εχουν απομεινει στον παιχτη.

3. `boolean played;` Μια boolean μεταβλητη που δειχνει εαν ο παιχτης εχει παιξει / ξεκινήσει τον γυρο του.

4. `boolean finished;` Μια boolean μεταβλητη που δειχνει εαν ο παιχτης εχει τελειώσει τον γυρο του.

5. `private String name;` Μια μεταβλητη που αποθηκευει το ονομα του παιχτη .

6. `private int id;` Μια μεταβλητη που αποθηκευει την ταυτοτητα / αναγνωριστικο του παιχτη.

7. private int action; Μια μεταβλητή που αποθηκεύει την ενέργεια που θέλει να κάνει ο παίκτης σε αυτόν τον γυρο.

Πακέτο Board:

Σε αυτό το πακέτο περιέχονται όλες οι μέθοδοι - κλάσεις που είναι υπεύθυνες για την αρχικοποίηση του board του παιχνιδιού, αλλά και των ενεργειών που μπορεί να εκτελέσει κάποιος παίκτης στην σειρά του.

Η κλάση Board: Αυτή η κλάση είναι υπεύθυνη για την αρχικοποίηση του board και την ανανέωση του μέχρι το τέλος του παιχνιδιού. Επίσης περιέχει μεθόδους για την σωστή κατεμετρηση των ποντών κάθε παίκτη, αλλά και για τον τερματισμό του παιχνιδιού.

Οι μέθοδοι της Board είναι:

1. public void addMosaicTile(MosaicTile mosaic); Αυτός είναι ένας Observer, ο οποίος προσθέτει το MosaicTile “mosaic” στο Mosaic_space.

2. public void addStatueTile(StatueTile statue); Αυτός είναι ένας Observer, ο οποίος προσθέτει το StatueTile “statue” στο Statue_space.

3. public void addAmphoraTile(AmphoraTile amphora); Αυτός είναι ένας Observer, ο οποίος προσθέτει το AmphoraTile “amphora” στο Amphora_space.

4. public void addSkeletonTile(SkeletonTile skeleton); Αυτός είναι ένας Observer, ο οποίος προσθέτει το SkeletonTile “skeleton” στο Skeleton_space.

5. public void addLandslideTile(LandslideTile landslide); Αυτός είναι ένας Observer, ο οποίος προσθέτει το LandslideTile “landslide” στο Entrance_space.

6. public void PlaceTiles(); Αυτός είναι ένας Transformer, ο οποίος τοποθετεί τα πρώτα 4 Tiles από την σακούλα στην αντίστοιχη θέση τους, για την αρχή του παιχνιδιού.

7. public void initializePlayersCharacters(Player p, String color); Αυτός είναι ένας Transformer, ο οποίος αρχικοποιεί όλες τις κάρτες χαρακτήρων του παίκτη “p”.

8. public boolean isChamberClosed(); Αυτός είναι ένας Observer, ο οποίος επιστρέφει true εάν ο χώρος των landslideTile είναι γεμάτος, αλλιώς επιστρέφει false.

9. `public int PointCount(Player p);` Αυτος ειναι ενας Accessor, ο οποίος επιστρεφει τους ποντους που εχει συγκεντρωσει ο παικτης “p” μετα το τελος του παιχνιδιου.

10. `public Board();` Αυτος ειναι ο constructor της Board. Δουλεια του ειναι η δημιουργια ενος καινουργιου Board object, αλλα και η αρχικοποιηση των λιστων: Mosaic_space, Statue_space, Amphora_space, Skeleton_space, Entrance_space.

Τα attributes της Board ειναι:

1. `private List <MosaicTile> Mosaic_space;` Αυτη ειναι μια λιστα που αποθηκευει ολα τα MosaicTiles που βρισκονται “πανω” στο board.

2. `private List <StatueTile> Statue_space;` Αυτη ειναι μια λιστα που αποθηκευει ολα τα StatueTiles που βρισκονται “πανω” στο board.

3. `private List <AmphoraTile> Amphora_space;` Αυτη ειναι μια λιστα που αποθηκευει ολα τα AmphoraTiles που βρισκονται “πανω” στο board.

4. `private List <SkeletonTile> Skeleton_space;` Αυτη ειναι μια λιστα που αποθηκευει ολα τα SkeletonTiles που βρισκονται “πανω” στο board.

5. `private List <LandslideTile> Entrance_space;` Αυτη ειναι μια λιστα που αποθηκευει ολα τα LandslideTiles που βρισκονται “πανω” στο board.

Η κλαση Turn: Αυτη η κλαση ειναι υπευθυνη για την δημιουργια ενος καινουργιου γυρου του παιχνιδιου, αλλα και για ολες τις πιθανες κινήσεις που μπορει να κανει ενας παικτης στον γυρο του.

Οι μεθοδοι της Turn ειναι:

1. `public Turn();` Αυτος ειναι ο constructor της Turn. Δουλεια του ειναι η δημιουργια ενος καινουργιου Turn object με id “id”.

2. `public void Remove4AndPlace(Board b);` Αυτος ειναι ενας Transformer, ο οποίος αφαιρει 4 πλακιδια στην τυχη απο την Bag και τα τοποθετει στο Board στις αντιστοιχες θεσεις.

3. `public void Take2FromBoard(List<Tile> PlayerTile);` Αυτος ειναι ενας Transformer, ο οποίος αφαιρει 2 πλακιδια απο το Board και τα τοποθετει στο stash του παιχτη.

4. `public void UseCharacter(List<Character> PlayerCharacters, Character character);` Αυτος ειναι ενας Transformer, ο οποίος αφαιρει τον χαρακτηρα "character" απο το stash του παιχτη και τον χρησιμοποιει.

5. `public int getTurn();` Αυτος ειναι ενας Accessor, ο οποίος επιστρεφει τον γυρο στον οποιο βρισκετε το παιχνιδι αυτη την στιγμη.

Τα attributes της Turn ειναι:

1. `private int turn;` Μια int μεταβλητη που αποθηκευει τον γυρο στον οποιο βρισκετε το παισινιδι αυτη την στιγμη.

Πακετο Characters: Σε αυτο το πακετο περιεχονται ολες οι μεθοδοι - κλασεις που ειναι υπευθυνες για την αρχικοποιηση ολων των καρτων χαρακτηρων του παιχνιδιου.

Η abstract κλαση Character: Η κλαση αυτη υπαρχει για την ομαδοποιηση των Character παρομοιου τυπου. Για παραδειγμα, extend αυτης της κλασης ειναι τα Archaeologist, Assistant, Digger και Professor.

Οι μεθοδοι της Character ειναι:

1. `public Character(int id, String name, String color);` Αυτος ειναι ο constructor της Character. Δουλεια του ειναι η δημιουργια ενος καινουργιου Character object με `belongs_to "id"`, `name "name"` και `color "color"`. Επισης θετει το `used` σε false.

2. `public void setCardHolder(int id);` Αυτος ειναι ενας Transformer, ο οποίος αρχικοποιει την `belongs_to` μεταβλητη σε "id".

3. `public void setColor(String color);` Αυτος ειναι ενας Transformer, ο οποίος αρχικοποιει την `color` μεταβλητη σε "color".

4. `public void setName(String name);` Αυτος ειναι ενας Transformer, ο οποίος αρχικοποιει την `name` μεταβλητη σε "name".

5. `public void changeUsed(boolean b);` Αυτος ειναι ενας Transformer, ο οποίος αλλαζει την τιμη της `used` μεταβλητης σε "b".

Τα attributes της Character είναι:

1. private boolean used; Μια boolean μεταβλητή που παίρνει την τιμή true εάν μια κάρτα έχει χρησιμοποιηθεί ή false εάν δεν έχει.
2. private int belongs_to; Μια int μεταβλητή που αποθηκεύει το id του παίκτη που έχει στην κατοχή του την κάρτα.
3. private String name; Μια μεταβλητή που αποθηκεύει το όνομα της κάρτας.
4. private String color; Μια μεταβλητή που αποθηκεύει το χρώμα της κάρτας.

Οι κλάσεις Archaeologist, Assistant, Digger, Professor: Οι κλάσεις αυτές όλες κάνουν extend την Character κλάση που αναφέρθηκε παραπάνω όπως φαίνεται και στο σχήμα 4. Το μόνο που έχουν είναι ένας constructor ο οποίος καλεί τον constructor της Character με belongs_to "id", name με Archaeologist, Assistant, Digger και Professor αντίστοιχα και color "color".

- Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Πακέτο Control: Αυτό το πακέτο είναι η καρδιά του παιχνιδιού. Εδώ δημιουργούνται όλοι οι παίκτες, οι κάρτες, τα Tiles, το Board και οι γυροί

Η κλάση Control:

Οι μέθοδοι της Control είναι:

1. public Controller(); Αυτός είναι ο constructor της Control. Δουλειά του είναι η δημιουργία ενός καινούργιου Control object και η αρχικοποίηση όλων των παικτών, της Bag, του Board και των μεταβλητών hasRemoved4, hasTaken2, hasPlayedCharacter, hasFinished και PlayerCount.
2. public void Start(); Αυτός είναι ένας Transformer, ο οποίος ξεκινάει το παιχνίδι και κάνει όλες τις απαραίτητες ενέργειες που απαιτούνται μέχρι το τέλος του παιχνιδιού.

3. `public boolean isSealed();` Αυτος ειναι ενας Observer, ο οποιος επιστρεφει true εαν η εισοδος ειναι σφραγισμενη, αλλιως επιστρεφει false.

4. `public void nextTurn(int id);` Αυτος ειναι ενας Transformer, ο οποιος “ξεκιναει” εναν καινουργιο γυρο. Το “id” δειχνει ποιος παιχτης θα παιξει αυτον τον γυρο.

Τα attributes της Control ειναι:

1. `private Player P1, P2, P3, P4;` Αυτα ειναι ολα τα object των παιχτων του παιχνιδιου.

2. `private ArrayList <Player> players;` Αυτη ειναι μια λιστα που αποθηκευει ολους τους παιχτες του παιχνιδιου.

3. `private Bag bag;` Αυτο ειναι ενα object τυπου Bag.

4. `private Turn turn;` Αυτο ειναι ενα object τυπου Turn.

5. `private Board board;` Αυτο ειναι ενα object τυπου Board.

6. `private GraphicUI view;` Αυτο ειναι ενα object τυπου View.

7. `private boolean hasRemoved4;` Αυτη ειναι μια boolean μεταβλητη που περνει την τιμη true, εαν ο παικτης εχει κανει την πρωτη του κινηση. Αλλιως περνει την τιμη false.

8. `private boolean hasTaken2;` Αυτη ειναι μια boolean μεταβλητη που περνει την τιμη true, εαν ο παικτης εχει κανει την δευτερη του κινηση. Αλλιως περνει την τιμη false.

9. `private boolean hasPlayedCharacter;` Αυτη ειναι μια boolean μεταβλητη που περνει την τιμη true, εαν ο παικτης εχει κανει την τριτη του κινηση. Αλλιως περνει την τιμη false.

10. `private boolean hasFinished;` Αυτη ειναι μια boolean μεταβλητη που περνει την τιμη true, εαν ο παικτης εχει κανει την τελευταια του κινηση. Αλλιως περνει την τιμη false.

11. `private int PlayerCount;` Αυτη ειναι μια int μεταβλητη που αποθηκευει το πληθος των παικτων σε αυτο την παρτιδα.

- Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Πακετο View: Σε αυτο το πακετο συμπεριλαμβανονται ολες οι κλασεις / μεθοδοι για την δημιουργια των γραφικων του παιχνιδιου (GUI). Το πακετο View ειναι ανεξαρτητο απο τον υπολοιπο κωδικα, συμφωνα με το τροπο σχεδιασης MVC.

Η κλαση GraphicUI:

Οι μεθοδοι της GraphicUI ειναι:

1. public void GraphicUI(); Αυτος ειναι ο constructor της GraphicUI. Δουλεια του ειναι η δημιουργια ενος καινουργιου GraphicUI object και η αρχικοποιηση ολων των JPanel, JButton και JTextField.

Τα attributes της Control ειναι:

1. private JPanel basic_panel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο βασικο / κυριο / πρωτο Panel των γραφικων.
2. private JPanel stashSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο stashSpace των παικτων.
3. private JPanel skeletonSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο skeletonSpace στο Board.
4. private JPanel amphoraSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο amphoraSpace στο Board.
5. private JPanel statueSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο caryatidSpace στο Board.
6. private JPanel mosaicSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο mosaicSpace στο Board.
7. private JPanel LandslideSpacePanel; Αυτο ειναι ενα object JPanel, το οποιο διχνει στο LandslideSpace στο Board.

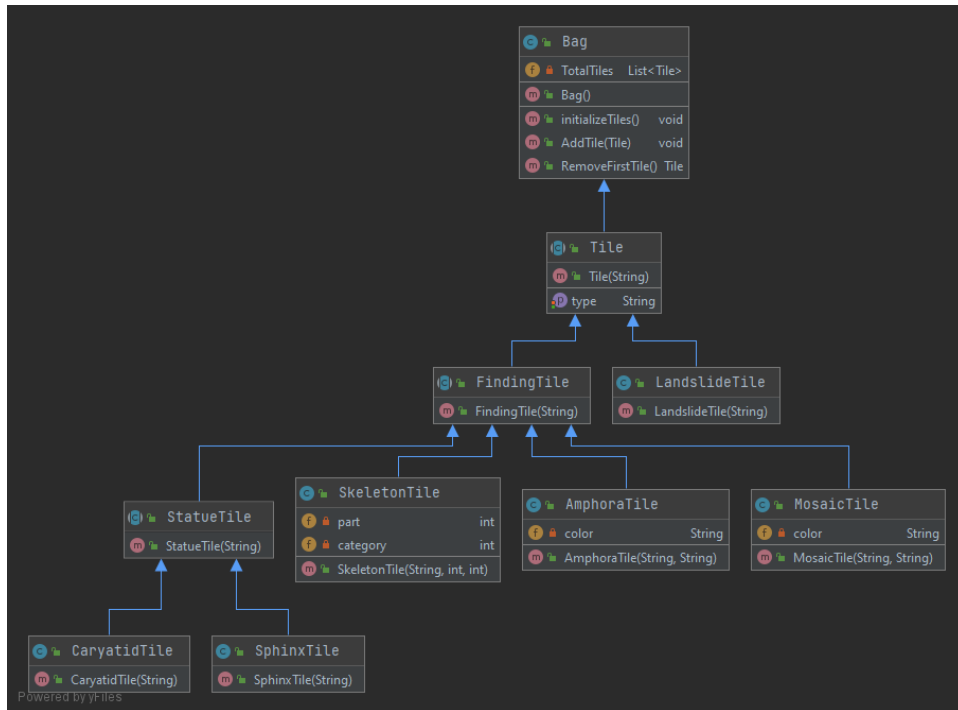
8. `private JPanel CharacterSpacePanel;` Αυτό είναι ένα object `JPanel`, το οποίο δίνει στο `CharacterSpace` στο `Board`.
9. `private JButton skeletonSpaceButton;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `skeletonSpaceButton` Button του `DrawTiles`.
10. `private JButton mosaicSpaceButton;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `mosaic SpaceButton` Button του `DrawTiles`.
11. `private JButton statueSpaceButton;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `statue SpaceButton` Button του `DrawTiles`.
12. `private JButton amphoraSpaceButton;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `amphoraSpaceButton` Button του `DrawTiles`.
13. `private JButton DrawTiles;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `DrawTiles Button` στο `Board`
14. `private JButton EndTurn;` Αυτό είναι ένα object `JButton`, το οποίο δίνει στο `EndTurn Button` στο `Board`.
15. `private JTextField playerPlaying;` Αυτό είναι ένα object `JTextField`, το οποίο θα δίνει το όνομα του παίκτη που παίζει αυτό τον γύρο στο `Board`.
16. `private JTextField Characters;` Αυτό είναι ένα object `JTextField`, το οποίο θα δίνει την λέξη `Characters` στο `Board`.

Η κλάση DrawTilesListener: Η κλάση αυτή κάνει implement την `ActionListener` και ο σκοπός της είναι η αναγνώριση του κάθε click που γίνεται στο `DrawTiles Button`. Κάθε φορά που γίνεται αυτό, ένα καινούργιο `JPanel` θα εμφανίζεται με 4 κουμπιά (παρακάτω εξηγώ ποια είναι αυτά και τις λειτουργίες τους) τα οποία θα επιτρέπουν στους παίκτες να “παρουν” τα `Tiles` που θέλουν από το `Board` στην σειρά τους.

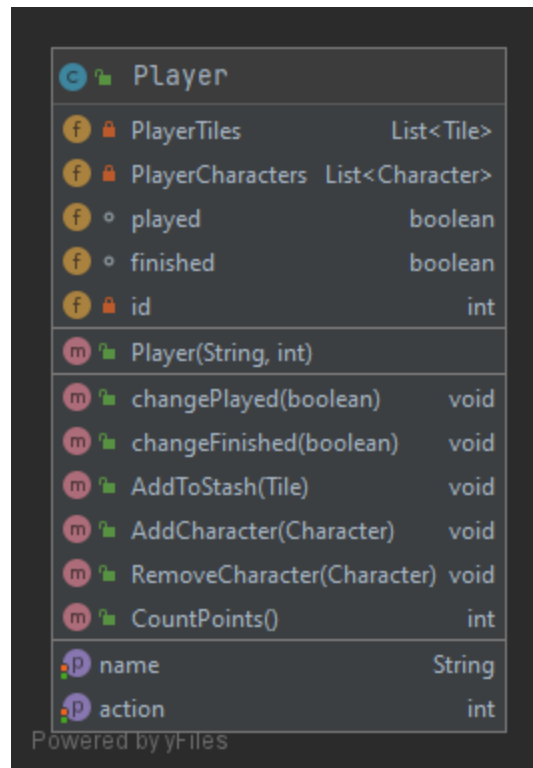
Οι κλάσεις skeletonSpaceListener, amphoraSpaceListener, statueSpaceListener, mosaicSpaceListener: Οι κλάσεις αυτές κάνουν implement την `ActionListener` και ο σκοπός τους είναι η αναγνώριση του κάθε click σε κάθε ένα από τα `Buttons` που τους αντιστοιχούν. Όταν αναγνωριστεί ένα click σε αυτά θα αφαιρείται ένα `Tile` από το αντιστοίχο `SpacePanel` τους και θα μεταφέρεται στο `StashSpacePanel` του παίκτη.

Η κλάση EndTurnListener: Η κλάση αυτή κάνει implement την ActionListener και ο σκοπός της είναι η αναγνώριση του κάθε click που γίνεται στο EndTurn Button. Κάθε φορά που γίνεται αυτό, ο παίκτης που πατήσε το κουμπί θα τελειώνει τον γύρο του, ξεκινώντας τον γύρο του επομένου παίχτη.

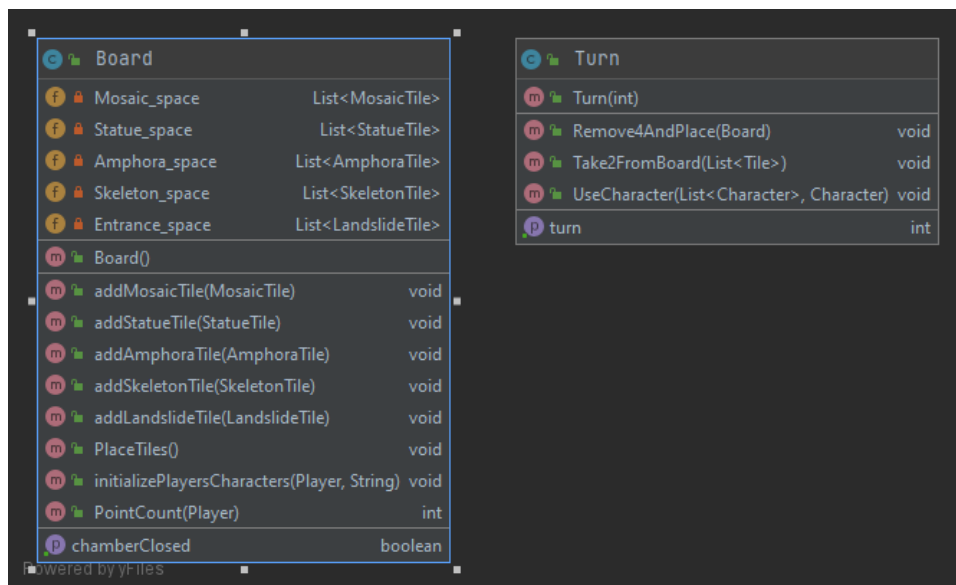
- Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



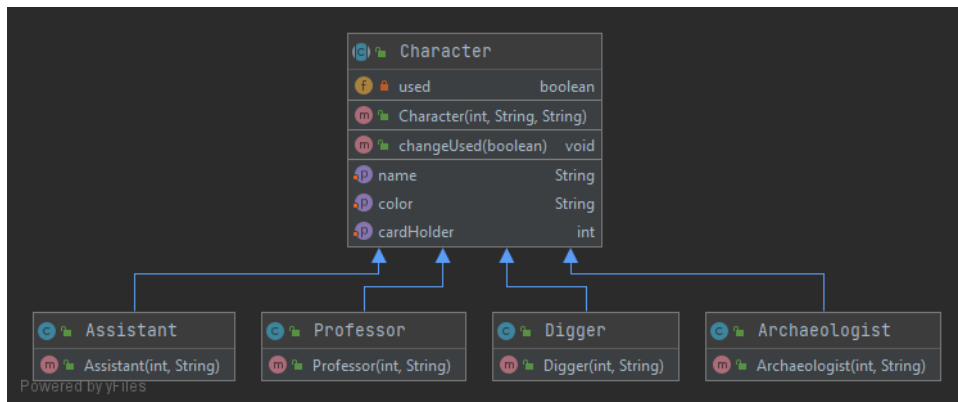
Σημα 1. UML διαγράμμα τους πακετου Tile.



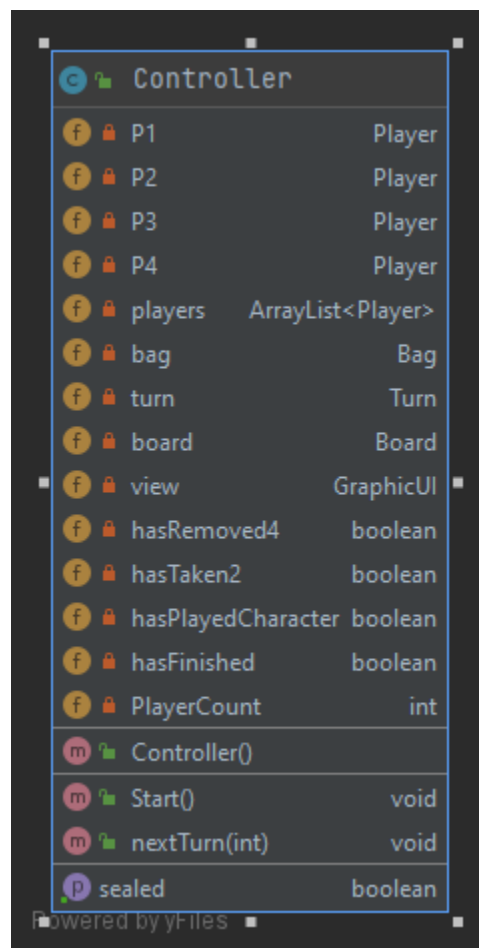
Σχήμα 2. UML διαγράμμα του πακέτου Player.



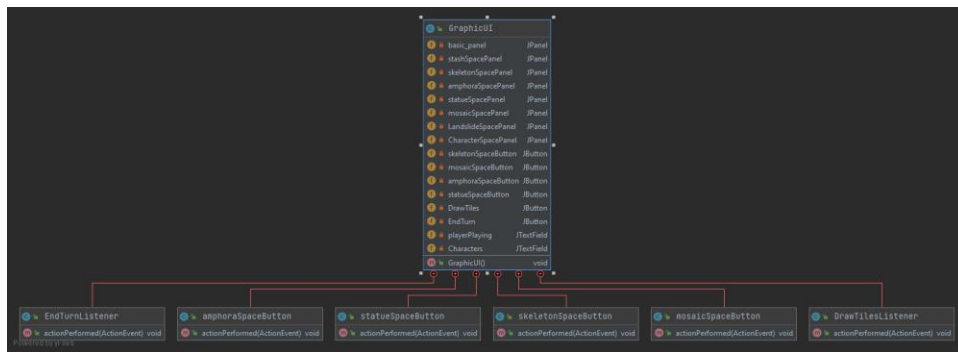
Σχήμα 3. UML διαγράμμα πακέτου Board.



Σχήμα 4. UML διαγράμμα πακέτου Character.



Σχήμα 5. UML διαγράμμα πακέτου Controller.



Σχημα 6. UML διαγραμμα πακετου View

- Λειτουργικότητα (B Φάση)

Σε αυτήν την ενότητα θα γράψετε στη B φάση ποια ερωτήματα καταφέρατε να υλοποιήσετε είτε επιτυχώς είτε εν μέρει (και ενδεχομένως ποια όχι).

- Συμπεράσματα

Καποιες λεπτομεριες σχετικα με το πακετο View και γενικότερα με το GUI δεν εχουν αναφερθει γιατι δεν ειμαι ακομη σιγουρος για το πως θα υλοποιησω το GUI (πως θα φαινεται). Εχω γραψει καποια βασικα στοιχεια του, που πιστευω πως θα υπαρχουν και στην τελικη υλοποιηση.