



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

AMPHIPOLIS

Κων/νος Ζιούτος

AM 3946

12/1/2021

Εισαγωγή

Think and describe what you plan to do and why it will be useful.

Περιεχόμενα

[ΟΒΥ]

[ΟΒΥ]

[ΟΒΥ]

[ΟΒΥ]

[ΟΒΥ]

[ΟΒΥ]

[ΟΒΥ]

1. Εισαγωγή

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller). Έτσι, σκοπός μας είναι ο Controller να είναι ο συνδετικός κρίκος των Model και view. Οπότε στη συνέχεια της αναφοράς μας θα αναλύσουμε λίγο ιδιαίτερα τα κομμάτια του Model και Controller που είναι σημαντικά για αυτή τη φάση και τέλος θα αναφερθούμε και λίγο στο view.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Σε αυτό το πακέτο υπάρχει η αφηρημένη κλάση Character οι υποκλάσεις της Archaeologist, Assistant, Digger, Professor. Επίσης υπάρχει η αφηρημένη κλάση Tile η αφηρημένη FindingTile που επεκτείνει την Tile και η αφηρημένη StatueTile που υλοποιεί την FindingTile. Οι υποκλάσεις της StatueTile οι CaryadTile, SphinxTile. Οι υποκλάσεις της FindingTile οι AmphoraTile, MosaicTile, SkeletonTile και η υποκλάση της Tile η LandslideTile. Ακόμα υπάρχουν οι enum κλάσεις AmphoraColor, FindingTileCategory, MosaicColor, SkeletonType, StatueType, TileCategory. Τέλος υπάρχει η διεπαφή GamePlayer, η κλάση Player που την υλοποιεί και οι κλάσεις bag και board

Πακέτο model.Cards

Character abstract and other card classes

Η abstract μας παρέχει τις εξής μεθόδους & attributes:

Attributes

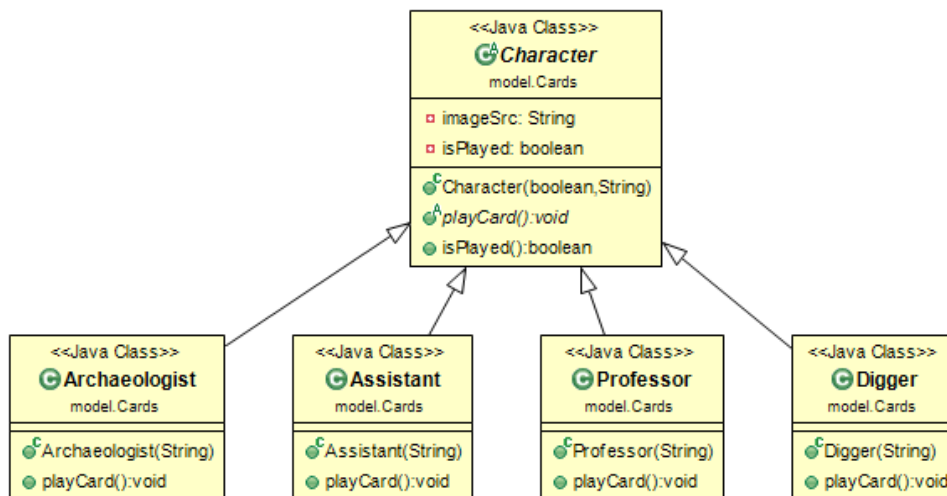
- **private String imageSrc** το string για την εικόνα
- **private boolean isPlayed**; μία μεταβλητή για το αν έχει παιχτεί

Methods

- **public abstract void playCard()**; Transformer.τι γίνεται κάθε φορά που παίζεται η κάθε κάρτα
- **public boolean isPlayed()** Observer.αν έχει παιχτεί ή όχι

Οι **Archaeologist,Assistant,Digger,Professor** επεκτείνουν την character και μέσω της super αποκτούν πρόσβαση σε αυτή και αρχικοποιούν τα παραπάνω attributes.

Τέλος θα δείξουμε μια αναπαράσταση των κλάσεων που έχουν σχέση με τις κάρτες μέσω UML



Πακέτο model.Tiles

Class Tile

Η κλάση αυτή είναι υπεύνηγη για την μοντελοποίηση κάθε πλακιδίου του παιχνιδιού

Attributes

- **private TileCategory tileCategory** η κατηγορία του πλακιδίου
- **private String tileImageSrc**; η εικόνα

Methods

- **public Tile(TileCategory tileCategory)** //constructor για τη κατηγορία finding χωρίς τη φωτογραφία που ορίζεται σε άλλο σημείο
- **public Tile(TileCategory tileCategory, String tileImageSrc)** // constructor για τη κατηγορία Landslide
- **public TileCategory getTileCategory()** //Accessor επιστρέφει τη κατηγορία
- **public String getTileImageSrc()** // Accessor επιστρέφει την εικόνα του πλακιδίου
- **public void setTileImageSrc(String tileImageSrc)** // transformer δίνουμε την εικόνα του πλακιδίου

Η **FindingTile** και η **LandslideTile** επεκτείνουν την Tile και μέσω super αποκτούν πρόσβαση στα attributes της. Επίσης η FindingTile έχει μία ακόμα μέθοδο την :

- **public FindingTileCategory getFindingTileCategory()** //accessor που επιστρέφει την κατηγορία των Finding πλακιδίων.

Η enum **TileCategory** έχει τις κατηγορίες των πλακιδίων.

Οι **MosaicTile, SkeletonTille, AmphoraTille** και η abstract **StatueTille** επεκτείνουν την FindingTile.

Η **MosaicTile** έχει ακόμα μέθοδο την

- **public MosaicColor getColor()** // accessor επιστρέφει το χρώμα

Το οποίο χρώμα το βρίσκουμε στην enum **MosaicColor**.

Η **SkeletonTille** έχει την

- **public SkeletonType getType()** //accessor επιστρέφει τον τύπο

Τον οποίο τύπο τον βρίσκουμε στην enum **StatueType**

Η **AmphoraTile** έχει την

- **public AmphoraColor getColor()** //accessor επιστρέφει το χρώμα

Το οποίο χρώμα το βρίσκουμε στην enum **AmphoraColor**

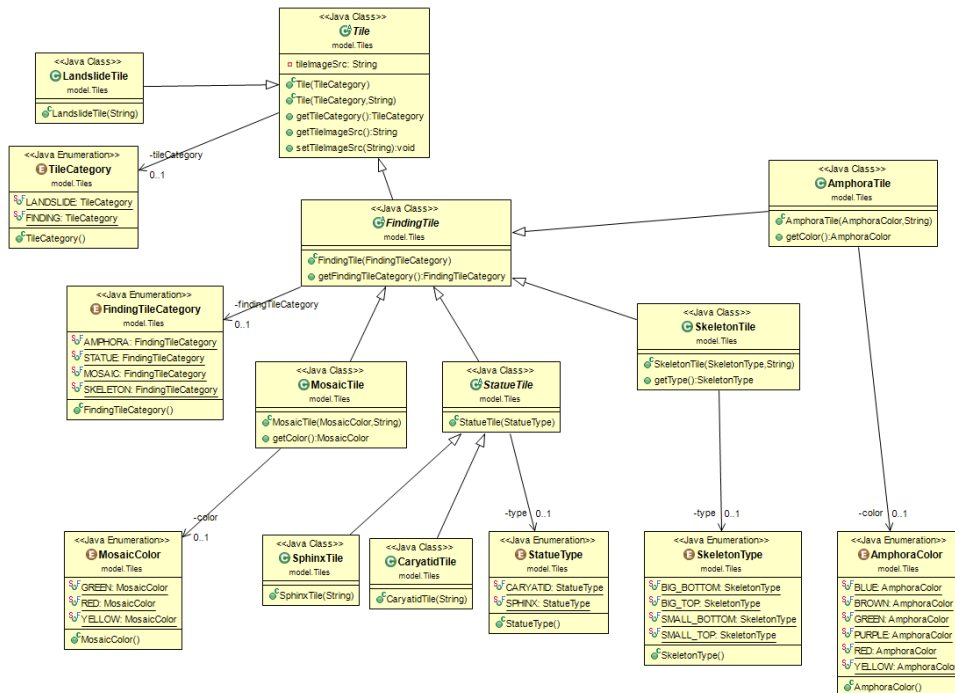
Η **StatueTile** έχει τον constructor

- **public StatueTile(StatueType type)**

Που περνάει μέσα τον τύπο τον οποίο τύπο τον παίρνει από την **enum StatueType**.

Οι **CaryatidTile**, **SphinxTile** επεκτείνουν την **StatueTile** και μέσω Super έχουν πρόσβαση.

Τέλος θα δείξουμε μια αναπαράσταση των κλάσεων που έχουν σχέση με τα πλακίδια μέσω UML.



Η Interface **GamePlayer** το οποίο περιέχει τις βασικές λειτουργίες του που έχει ο κάθε παίχτης του παιχνιδιού δηλαδή τι μπορεί να κάνει ένας παίχτης στο γύρο του περιέχει τις ακόλουθους μεθόδους :

- **public String getPlayerName()** //accessor επιστρέφει το όνομα του παίχτη
- **public void getBoardTiles(ArrayList<Tile> tiles)** //accessor παίρνει τα πλακίδια από την περιοχή επιλογής και τα αποθηκεύει στη λίστα με τα πλακίδια
- **public ArrayList<Tile> returnFindingTiles()** //transformer επιστρέφει τα πλακίδια που έχει συλλέξει ο παίχτης.
- **public void playCharacter(Character card)** //transformer παίζει μια κάρτα χαρακτήρα από το χέρι και παίρνει τα πλακίδια που πρέπει.

Η κλάση **Player** μοντελοποιεί τον παίχτη του παιχνιδιού έχει τα παρακάτω attributes:

- **String playerName** //όνομα παίχτη
- **private ArrayList<Tile> earnedTiles** //τα πλακίδια που έχει συλλέξει ο παίχτης
- **private ArrayList<CharacterCard> characterCards** //οι διαθέσιμες κάρτες που έχει ένας παίχτης
- **private int playerId** //για την καταμέτρηση
- **private int gottenTilesForTurn;** //πόσα tiles πήρε στο γύρο
- **private FindingTileCategory selectedTileAreaForThisTurn;** //επιλεγμένα tiles

Methods

- **public Player(int index, String name)**// constructor
- **public void addCards(CharacterCard card)**//transformer προσθέτει κάρτες
- **public ArrayList<CharacterCard> getCharacterCards()** //accessor παίρνει κάρτα χαρακτήρων
- **public String getPlayerName()**//accessor επιστρέφει το όνομα του παίχτη
- **public void addTileToEarnedTiles(Tile tile)** //transformer προσθέτει το κομμάτι που επέλεξε ο παίχτης στη λίστα του

- **public ArrayList<Tile> getEarnedTiles()** // accessor επιστέφει τα κερδισμένα tiles
- **public boolean isPossibleToGetThisTile(FindingTileCategory currentTileArea)** //observer έλεγχος αν μπορεί να τραβήξει από μια περιοχή ή όχι
- **public void preparePlayerToStartHisTurn()** //transformer καλούμε κάθε φορά που τελειώνει ο γύρος ενός παίχτη πριν πάμε στον άλλον
- **public int getAmphoraScore()** //accessor υπολογίζει σκορ των αμφορέων
- **public int getSkeletonsScore()** // accessor υπολογίζει σκορ των σκελετών
- **public int getMosaicsScore()** // accessor υπολογίζει σκορ των μωσαϊκών
- **public int getSphinxNum()** //accessor υπολογίζει αριθμό σφίγγων
- **public int getCaryatidNum()** // accessor υπολογίζει αριθμό Καρυάτιδων
- **public int getStatueScore(int caryatidNum1, int caryatidNum2, int caryatidNum3, int sphinxNum1, int sphinxNum2, int sphinxNum3)** // accessor υπολογίζει σκορ των αγαλμάτων

Η κλάση **Bag**, μοντελοποιεί τη σακούλα με τα πλακίδια του παιχνιδιού

Attributes

- **private ArrayList<Tile> bagTiles** // λίστα με τα πλακίδια που έχουν απομείνει στη σακούλα

Methods

- **public Bag(ArrayList<Tile> listTiles)** //constructor
- **public Tile getFistAmphoraTile()** //accessor επιστρέφει ένα τυχαίο πλακίδιο αμφορεά για την τοποθέτηση κατά την αρχικοποίηση
- **public Tile getFirstMosaicTile()** //accessor αντίστοιχα για μωσαϊκό
- **public Tile getFirstStatueTile()** //accessor αντίστοιχα για άγαλμα
- **public Tile getFirstSkeletonTile()** // accessor αντίστοιχα για σκελετό
- **public ArrayList<Tile> getRandomTile()** // accessor επιστρέφει ένα πλακίδιο τυχαία οποιαδήποτε τύπου τα οποία έχει αφαιρέσει από τη λίστα

Η κλάση **Board** είναι υπεύθυνη για τη μοντελοποίηση του ταμπλό του παιχνιδιού

Attributes

- **private ArrayList<CharacterCard> cardsForPlayer1;** //καρτες παιχτών

- **private ArrayList<CharacterCard> cardsForPlayer2;**
- **private ArrayList<CharacterCard> cardsForPlayer3;**
- **private ArrayList<CharacterCard> cardsForPlayer4;**
- **private ArrayList<Tile> allGameTiles;** //τα πλακίδια για την αρχικοποίηση
- **private ArrayList<Tile> boardTiles;** //τα πλακίδια πάνω στο ταμπλό

Methods

- **public Board()** // constructor
- **public void initGameCards ()** //transformer αρχικοποιεί τις κάρτες χαρακτήρων
- **public void initGameTiles ()** //transformer αρχικοποιεί τα tiles
- **public ArrayList<CharacterCard> getPlayer1Cards()** // accessor παίρνει τις κάρτες χαρακτήρων για παίχτη 1
- **public ArrayList<CharacterCard> getPlayer2Cards()** // accessor παίρνει τις κάρτες χαρακτήρων για παίχτη 2
- **public ArrayList<CharacterCard> getPlayer3Cards()** // accessor παίρνει τις κάρτες χαρακτήρων για παίχτη 3
- **public ArrayList<CharacterCard> getPlayer4Cards()** // accessor παίρνει τις κάρτες χαρακτήρων για παίχτη 4
- **public ArrayList<Tile> getAllGameTiles()** //accessor παίρνει όλα τα tiles του παιχνιδιού
- **public ArrayList<Tile> getBoardTiles()** //accessor παίρνει τα tiles του ταμπλό
- **private void initAmphoraTiles()** //transformer αρχικοποίηση αμφορέων
- **private void initMosaicTiles()** // transformer αρχικοποίηση μωσαϊκών
- **private void initStatueTiles()** // transformer αρχικοποίηση αγαλμάτων
- **private void initLandslideTiles()** // transformer αρχικοποίηση πλακιδίων κατολίσθησης
- **private void initSkeletonTiles()** // transformer αρχικοποίηση σκελετών

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Αυτή η κλάση είναι το μυαλό του παιχνιδιού. Δημιουργεί στιγμιότυπα των παιχτών των καρτών του ταμπλό. 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. Και είναι υπεύθυνη να βρει το νικητή

Attributes

- **private Board board;** // ταμπλό
- **private Bag bag;** //σάκος
- **private View view;** // η σύνδεση με τη View
- **Player players[] = new Player[4];** // οι 4 παίκτες
- **int activePlayerIndex;** //το ιντεξ του παίχτη

Methods

- **public void initialize()** //transformer αρχικοποίηση παιχνιδιού
- **public void setListeners()** //transformer θέτει τα listeners
- **private class EndTurnListener implements ActionListener** //transformer τι κάνει στο τέλος του γύρου
- **private class DrawTilesListener implements ActionListener** //transformer τι γίνεται όταν τραβάμε πλακίδια
- **public void onMosaicMouseClicked(MouseEvent e)** //transformer όλες οι ενέργειες για να παέι το mosaic tile από το board στο παίχτη και ότι χρειάζεται για να γίνει η απεικόνισή στο view
- **public void onStatueMouseClicked(MouseEvent e)** // transformer όλες οι ενέργειες για να πάει το statue tile από το board στο παίχτη και ότι χρειάζεται για να γίνει η απεικόνισή στο view
- **public void onSkeletonMouseClicked(MouseEvent e)** // transformer όλες οι ενέργειες για να πάει το skeleton tile από το board στο παίχτη και ότι χρειάζεται για να γίνει η απεικόνιση στο view
- **public void onAmphoraMouseClicked(MouseEvent e)** // transformer όλες οι ενέργειες για να παέι το amphora tile από το board στο παίχτη και ότι χρειάζεται για να γίνει η απεικόνιση στο view
- **private class CardListener implements MouseListener** //transformer όταν πατάμε στη κάρτα
- **private void playersGetTheirCards()** //transformer παίκτες παίρνουν τις κάρτες τους
- **private void set4FirstTilesToBoard()** //transformer βάζουμε 4 πλακίδια στην αρχή στο ταμπλό

- **private boolean checkIfGameFinished(ArrayList<Tile> boardTiles)**
//observer τσεκάρει αν το παιχνίδι έχει τελειώσει
- **private Tile getFindingTileFromBoard(FindingTileCategory tileCategory, int index)** // accessor παίρνει το πλακίδιο που επέλεξε ο χρήστης και το επιστρέφει

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View περιέχει την κλάση View η οποία είναι υπεύθυνη για την απεικόνιση του παιχνιδιού. Η σχεδίαση που επιλέχθηκε είναι αντίστοιχη αυτής της εκφώνησης. Η κλάση View θα δημιουργεί ένα Frame, το οποίο θα έχει 1 πάνελ για την κάθε περιοχή με πλακίδια, καθώς και ένα για τα πλακίδια κατολισθησης. Κάθε ένα πλακίδιο από τις περιοχές των ευρημάτων θα είναι και ένα button το οποίο θα μπορεί να πατήσει ο χρήστης και να το επιλέξει. Η περιοχή με τις κάρτες των χαρακτήρων θα δημιουργούνται με αντίστοιχο τρόπο.

Θα υπάρχουν ακόμα τα buttons “Draw Tiles” και “End Turn” τα οποία κατά αντιστοιχία: θα οριστικοποιούν τις επιλογές σε πλακίδια που θα κάνει ο χρήστης και θα ολοκληρώνουν τον γύρο του παίχτη.

Attributes

- **private JButton[] playerCards = new JButton[4];** //κάρτες χαρακτήρων κουμπιά
- **private JLabel[] mosaicPositions;** //Jlabels για τις θέσεις στο ταμπλό
- **private JLabel[] statuesPositions;**
- **private JLabel[] amphoraPositions;**
- **private JLabel[] skeletonsPositions;**
- **private JLabel[] landslidePositions;**
- **private JLabel[] playerTilesPositions;**
- **private view.JLayeredPaneExtension basic_panel;** //το βασικό πάνελ
-
- **private JLayeredPane cardsPane;** //πανελ καρτών
- **private JLayeredPane buttonsPane;** //πανελ κουμπιών
- **private ClassLoader cldr;**
- **private JButton drawTilesButton;** //κουμπί για να τραβάμε πλακίδια
- **private JButton endTurnButton;** //κουμπί για να τελειώνει η παρτίδα
- **JLabel tile;**
- **JTextArea infobox;**

Methods

- **public JButton[] getPlayerCards()** //accessor κουμπί να παίρνει τους χαρακτήρες
- **public JLabel[] getMosaicPositions()** //accessor παίρνει τις θέσεις των μωσαϊκών
- **public JLabel[] getStatuesPositions()** //των αγαλμάτων
- **public JLabel[] getAmphoraPositions()** //των αμφοραίων
- **public JLabel[] getSkeletonsPositions()** //των σκελετών
- **public void initCards(ArrayList<CharacterCard> cards)** //αρχικοποίηση καρτών
- **public void initMainButtons()** // αρχικοποίηση κουμπιών
- **public View()** //constructor
- **public void addTilesToBoard(ArrayList<Tile> boardTiles)** // transformer βάζω πλακίδια στο πίνακα
- **public void addDrawTilesListener(ActionListener listenDrawTilesButton)** //trnsformer βάζω κουμπί για να τραβάει πλακίδια
- **public void addEndTurnListener(ActionListener listenerEndTurnButton)** //transformer βάζω κουμπί για να τελειώνει ο γύρος
- **public void initComponents(ArrayList<CharacterCard> cards,ArrayList<Tile> boardTiles)** //transformer αρχικοποιώ τα components
- **private void clearAllBoardViewsTiles()** //transformer καθαρίζει το ταμπλό
- **private void clearPlayerEarnedViewsTiles()** //transformer καθαρίζει τα κερδισμένα πλακίδια
- **public void updatePlayerCards(ArrayList<CharacterCard> playerCards)** //αναβαθμίζει τις κάρτες χαρακτήρων

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Σε αυτό το διάγραμμα βλέπουμε την αλληλεπίδραση μεταξύ των κλάσεων και την ιεραρχία τους. Πως επικοινωνούν μεταξύ τους οι κλάσεις. Βλέπουμε τις συνδέσεις μέσα στο model και επίσης επιβεβαιώνουμε ότι ο controller είναι ο εγκέφαλος του παιχνιδιού αφού επικοινωνεί τόσο με το model όσο και με το view.

