

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2020-2021**

AMPHIPOLIS

Εισαγωγή

Τζανάκης Νικόλαος

csd 4349

8/12/2020

***Project 2020***

Περιεχόμενα

[1. Εισαγωγή 1](#_Toc530045458)

[2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 1](#_Toc530045459)

[3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller 1](#_Toc530045460)

[4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View 2](#_Toc530045461)

[5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 2](#_Toc530045462)

[6. Λειτουργικότητα (Β Φάση) 2](#_Toc530045463)

[7. Συμπεράσματα 2](#_Toc530045464)

## Εισαγωγή-Σχεδιασμός της εργασίας

Η υλοποίηση της άσκησης θα πραγματοποιηθεί με το μοντέλο (MVC) ,δηλαδή θα χρησιμοποιηθεί ένας Controller ο οποίος ανάλογα με τις επιλογές του παίκτη θα στέλνει εντολές στα αντικείμενα Model,τα οποία περιέχουν πληροφορίες και διαχειρίζονται τα δεδομένα κατάλληλα,και στο view το οποίο ευθύνεται για την γραφική απεικόνιση πληροφοριών στους παίκτες.Στη συνέχεια της αναφοράς,θα αναφερθώ στην υλοποίηση των Model ,Controller και View με ιδιαίτερη προσοχή στο Model.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακέτο Model ευθύνεται για τα Μοντέλα του παιχνιδιού,δηλαδή για τα Tiles,

για τις καρτες χαρακτήρων,για την συλλογη των tiles ,για την διαχείρησει του παίκτη και για το board.

***Model.Tiles***

**Enum Colors**

Το Εnum Colors δημιουργήθηκε για να ορίσουμε ποια χρώματα θα χρησιμοποιηθούν σε αυτό το παιχνίδι.

**Abstract Class Tile**

Μετην abstract Class Tile,μπορω να ορίσω γενικές μεθόδους που θα χρειαστούν σε κάθε ξεχωριστό tile,και να αποθηκέυσω δεδομένα για αυτά.Μέθοδοι που αρχικά δίνονται:

public int getState()

είναι για να ελέγχω σε τι κατάσταση ειναι το tile,δηλαδή αν ειναι στη τσάντα,στο ταμπλό,ή το έχει κάποιος παίκτης

public void setState(int state)

για να ορίσω σε ποια κατάσταση βρίσκεται το tile

public String getCategory()

είναι για να ξέρω σε ποια κατηγορία ανήκει ενα συγκεκριμμένο tile.

Στη συνέχεια έχω την **LandSlideTile** class η οποια κανει extends την Class Tile

δεν έχει κάποια άλλη μέθοδο εκτος απο τον constructor.

**Abstract Class FindingTile**

Άλλη μια abstract class η οποία υλοποιεί τις μεθόδους που χρειάζονται τα tiles που δίνουν πόντους και αυτά που αναζητά κάθε παίκτης.Κάθε τέτοια κλάση <name>Tile κανει extend την κλάση FindingTile.Έφτιαξα την FindingTile για να ξεχωρίσω τα landSlideTiles με αυτά που μαζεύουν οι παίκτες

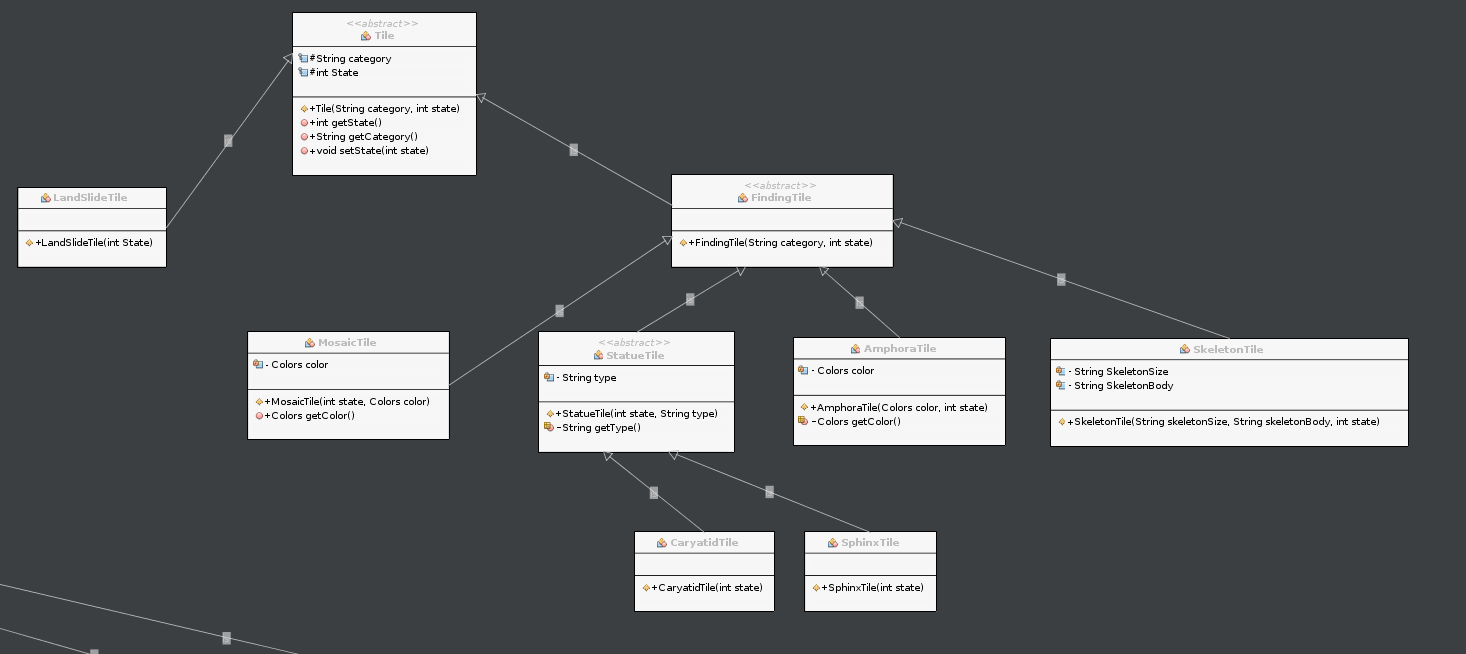
**Class AmphoraTile, MosaicTile, SkeletonTile**

Αυτές οι κλάσεις χρησιμεύουν για να φτιάχνονται αντικείμενα και για να ορίσω κάποιες ιδιαιτερότητες όπως για το χρώμα τον αμφορέων.Αυτά τα Tiles ειναι αυτά που μαζεύουν οι παίκτες, άρα κάνουν extend την FindingTile Class.

**Abstract Class StatueTile**

H κλάση StatueTile ειναι abstract,και κάνει extend την FindingTile Class, διότι έχουμε δύο κατηγορίες τέτοιου τύπου,το Sphinx και την Caryatid,οπότε την έκανα abstract και έφτιαξα άλλες δύο κλάσεις(Class SphinxTile kai Class CaryatidTile) οι οποίες κάνουν extend την StatueTile,για να φτιάχνω αντικείμενα Sphinx και Caryatid

Το διάγραμμα UML όπου φαίνεται ξεκάθαρα η υλοποίηση για τα tiles:



**Model.Cards**

Αυτό το πακέτο περιλαμβάνει την υλοποίηση για τις κάρτες χαρακτήρων που χρησιμοποιεί κάθε παίκτης.

**Abstract Class Card**

Αυτή η κλάση χρησιμοποιείται για την υλοποίηση των μεθόδων και τα χαρακτηριστικά των καρτών Χαρακτήρα.

boolean used.

Για να ξέρουμε αν η κάρτα εχει χρησιμοποιηθει στο παιχνίδι

String attribute.

Το χαρακτηριστικό της κάρτας Χαρακτήρα

Colors cardHolder.

Το Χρώμα του παίκτη που κρατάει αυτη την κάρτα.

Public boolean isAvailable()

μέθοδο για να ξέρουμε αν η κάρτα μπορει να χρησιμοποιηθεί.Γυρνάει την τιμή του used

public void setUsed().

Μέθοδο για να ορίσουμε οτι η κάρτα χρησιμοποιήθηκε

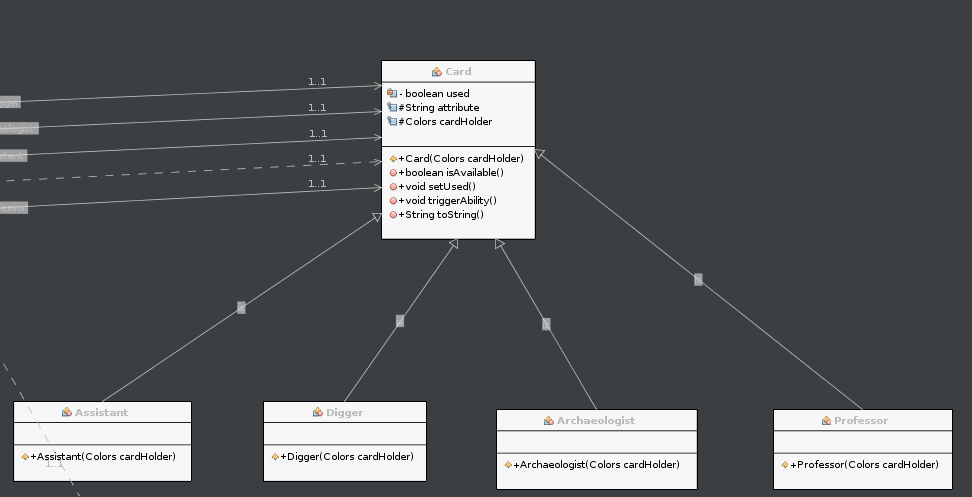
public void triggerAbility()

Μέθοδο για να πραγματοποιηθεί η ικανότητα της κάρτας.

**Class Archaeologist,Assistant,Digger,Professor**

Αυτές οι κλάσεις είναι για να δημιουργήσουμε αντικείμενα(τις καρτες) για τον κάθε παίκτη.Αρχικά περιέχουν τον constructor τους που ορίζουν το χαρακτηριστικό κάθε κάρτας.ολες κάνουν extend την Class Card,η οποία είναι υπεύθυνει για την λειτουργία τους.

Ορίστε ένα διάγραμμα UML:



**Model.Player**

**(Να τα φτιαξω πιο ομορφα για να ξεχωριζουν)**

Η κλάση Player εχει τις λειτουργιες και τις μεθόδους κάθε παίκτη.Επίσης έχει και τις κάρτες χαρακτήρα.Επιπλέον,έχει ενα collection απο tiles,για τα tiles που εχει μαζέψει στο παιχνίδι.Για να ξεχωριζουμε τους παίκτες κάθε παίκτης έχει ένα ξεχωριστό Color.Επίσης κάθε παίκτης έχει τους πόντους του.Οι βασικές μέθοδοι:

**public String getName()**

Επιστρέφει το όνομα του παίκτη.

**Public Colors getColor()**

Επιστρέφει το χρώμα του παίκτη .

**Public card getProfessor(),getDigger(),getAssistant(),getArchaeologist()**

Επιστρέφουν την κάρτα χαρακτήρα του παίκτη.

**Public void calculatePoints()**

Υπολογίζει τους πόντους του παίκτη

**public int getPoints()**

Επιστρέφει τους ποντους του παίκτη

**public void setEndTurn(boolean turn)**

Με αυτη τη μέθοδο ορίζουμε αν ο παίκτης έχει τελειώσει τον γύρο του ή όχι

**public boolean getEndTurn()**

Επιστρέφει την κατάσταστη του endTurn

***Model.Collection***

**Class Collection**

Η κλάση Collection υλοποιεί ότι εχει να κάνει με λιστες απο tiles.Περιέχει δηλαδή ένα arrayList απο Class <Tile> και έχει μεθόδους που επεξεργάζεται αυτό το arrayList.

**Public void initializeTiles()**

Αυτη η μέθοδο χρησιμοποιείται για να φτιάξει την τσάντα με ολα τα tiles που χρειαζονται στην αρχη.

**Public void addTile(Tile i)**

Προσθέτει το Tile I στο arrayList απο tiles.

**Public void removeTile(Tile I)**

αφαιρεί το Tile I απο την λίστα,αν αυτό υπάρχει

**public String getTileString(int I)**

επιστρέφει την toString() του tile στη θέση I

**public int getSize()**

επιστρέφει το μέγεθος του arrayList δηλαδη του collection

**public Tile getTile(int I)**

επιστρέφει το Tile στη θέση I,αν υπάρχει κάποιο tile στη θέση I

**public boolean isEmpty()**

ελέγχει αν το arrayList απο tiles ειναι άδειο

***Model.Board***

***Class Board***

Η κλάση board χρησιμοποιείται για τις λειτουργίες πάνω στο board.Συγκεκριμμένα,περιέχει τις περιοχές των tiles,τα οποία είναι collections απο tiles.

***Μέθοδοι:***

**public void addTile(Tile i)**

Η μέθοδος αυτή χρησιμοποιούνται για να προσθέσουμε ανάλογα το Tile στην ανάλογη Collection του board.

**Public void removeTile(Tile i)**

Αυτη η μέθοδο ειναι για να αφαιρέσουμε ένα tile απο κάποια Λίστα

**public int getTiles(Tile i)**

επιστρέφει το μέγεθος μιάς περιοχής του tile I.

**Public void StartingBoard()**

Αυτη η μέθοδος ειναι για την αρχικοποίηση του board οταν ξεκινάει το game.

**Public boolean endGame()**

Ελέγχει αν έχει τελείωσει το game.Δηλαδη αν υπάρχουν ολα τα landSlideTiles στην περιοχή με τα landSlides

**public Colors nextPlayer()**

επιστρέφει το χρώμα του παίκτη που παίζει τον επόμενο γύρο

**public boolean hasPlayer()**

Ελέγχει αν ο παίκτης έχει τελειώσει τον γύρο του

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

***Package Controller***

***Class Controller***

Ο controller χρησιμοποιεί ιδανικά το Model ώστε να συνδυάσει τα μοντέλα και να “τρέχει” το παιχνίδι.

**Χαρακτηριστικά:**

Στη κλαση Controller έχουμε τους 4 παίκτες (private Player P1,P2,P3,P4),ενα board(private Board board),ένα collection το οποίο είναι η σακούλα με τα tiles(private Collection bag)

Στον constructor της κλάσης Controller δημιουργόυνται οι παίκτες με τα χρώματα και τα ονόματα τους,ορίζει τον παίκτη που παίζει πρώτος,δηλαδή ο P1,φτιάχνει τα tiles της τσάντας και καλει την μέθοδο StartingBoard του board.

**Μέθοδοι:**

public boolean getHasPlayer()

Ελέγχει αν ο παίκτης έχει τελειώσει τις κινήσεις του.και επιστέφει την ανάλογη boolean τιμή.

Public Colors getCurrentPlayer()

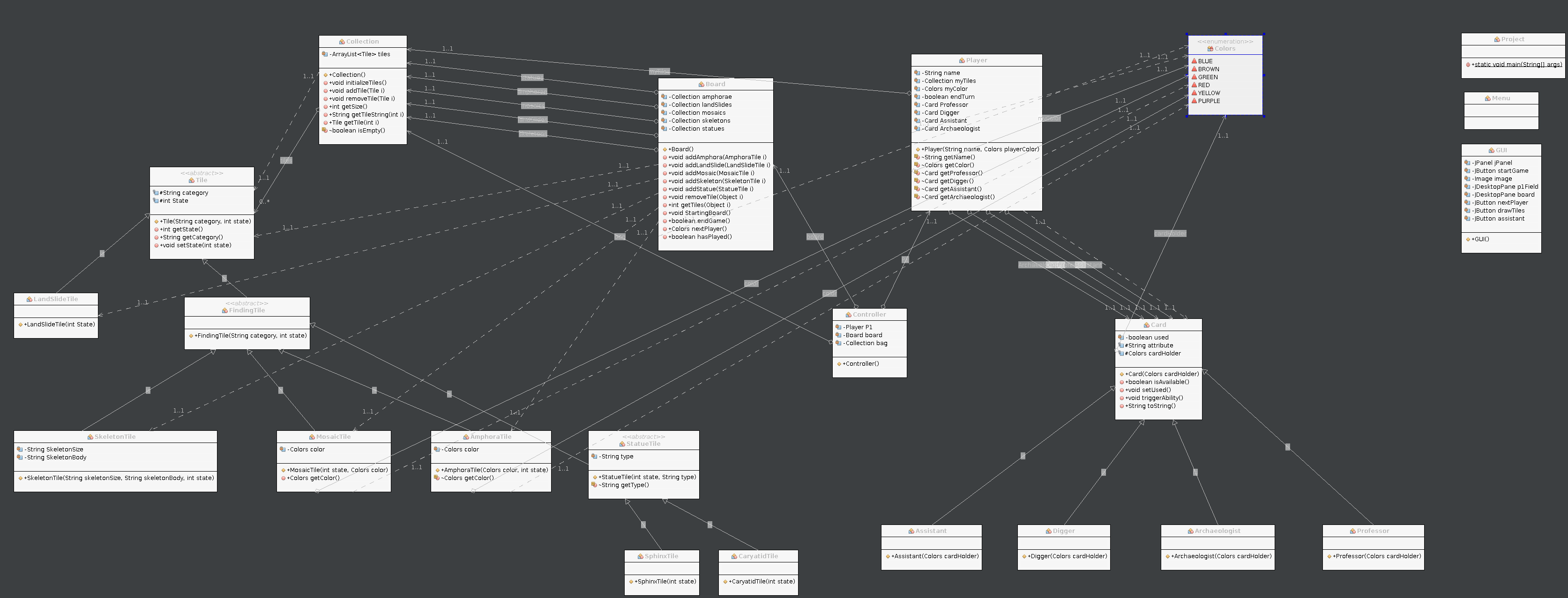
Επιστρέφει τον παίκτη που παίζει στον γύρο που παίζεται εκείνη την ώρα

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

***Package View***

Το package View περιέχει την GUI class,η οποία είναι υπεύθυνη για το γραφικό περιβάλλον την ώρα του παιχνιδιού,και μια κλάση Menu , η οποια ειναι γραφικη απεικόνιση πριν αρχίσει το παιχνίδι,η θα εμφανίζεται αν γίνει backToMenu την ώρα του παιχνιδιου.Γενικά δεν έχω γράψει πολλά στην Α φάση γιατί θέλω να τεστάρω στη β φάση τις εικόνες μαζι με το μέγεθος των παραθύρων. Αυτό που είναι σίγουρο ειναι οτι θα υπάρχουν ButtonListeners για τα tiles,αυτα δηλαδη που θα μαζευει καποιος παικτης στο board,και για τις καρτες χαρακτήρα,για το τράβιγμα των tiles,για το τέλος του γύρου,και θα υπάρχει και ενα setting για το αν θέλουμε να τερματισουμε το παιχνίδι,να το αρχισουμε απο την αρχή ή να επιστρέψουμε στο menu.Γενικά,κάθε παίκτης θα έχει το δικό του παράθυρο,δηλαδή μετά το τέλος του γύρου ενος παίκτη,θα αλλάζουν τα χρώματα και οι παίκτες χαρακτήρα,όπως και τα tiles που εχει μαζέψει.Τα αλλα νομίζω θα τα αφήσω τα ίδια.Ανάλογα και με το πως θα βολέψει στον κώδικα.

## Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Έχω αναφερθεί και στα διαγράμματα UML στις προηγούμενες ενότητες,αλλα εδώ παραθέτω το ολοκληρωτικό διάγραμμα. Βλέπουμε αριστερά οτι έχουμε την abstract class Tile,η οποία έχει 2 υποκλάσεις , την class LandSlideTile,και την abstract class FindingTile,η οποία FindingTile εχει 4 υποκλάσεις(τις κατηγορίες των tiles)όπου η statueTile ειναι και αυτη abstract,οπου εχει τις SphinxTile kai CaryatidTile ως υποκλάσεις.Στη κάτω δεξια μεριά φαίνεται η abstract Class Card,η οποία έχει ως υποκλάσεις τις κλάσεις των καρτών Χαρακτήρων.Πάνω αριστερά φαίνεται το Collection,οπου έχει ένα χαρακτηριστικό,ένα arrayList απο Objects Tile

Στο κέντρο έχουμε το board,οπου έχει χαρακτηριστικά τις περιοχές ,δηλαδή Collections.

Δεξιά απο το board έχουμε την κλάση Player,η οποία έχει χαρακτηριστικά απο Collection,Card kai Colors.Φαίνεται απο το διάγραμμα οτι έχει ως χαρακτηριστικά όλες τις κάρτες χαρακτήρα

Βλέπουμε και το enum Colors όπου “μοιράζει” χαρακτηριστικά σε άλλες κλάσεις

Και τέλος έχουμε και την Main οπου απλα θα καλεί το GUI,το menu οπου δεν το εχω φτιάξει στην Α φάση και το GUI οπου θα υλοποιηθει στην Β φάση

## Λειτουργικότητα (Β Φάση)

Ξεκινώντας,δημιουργησα την bag με τα tiles ,χωρις να ειναι σε τυχαια σειρά,δηλαδή ίδιες κατηγορίες είναι δίπλα δίπλα και απλα η τυχαιότητα θα πραγματοπείται με την random της Java.Στη συνέχεια αρχικοποιώ τα πρώτα Tiles στο board.

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

## Συμπεράσματα

Γενικά θα δημιουργηθούν και προβλήματα κατα την διάρκεια του κώδικα και της υλοποίησης της Β φάσης,οπότε ενδεχομένως να λείπουν αρκετοί μέθοδοι στις κλάσεις.

Νομίζω οτι τα πολύ βασικά είναι αυτα που έχω αναφέρει.