Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

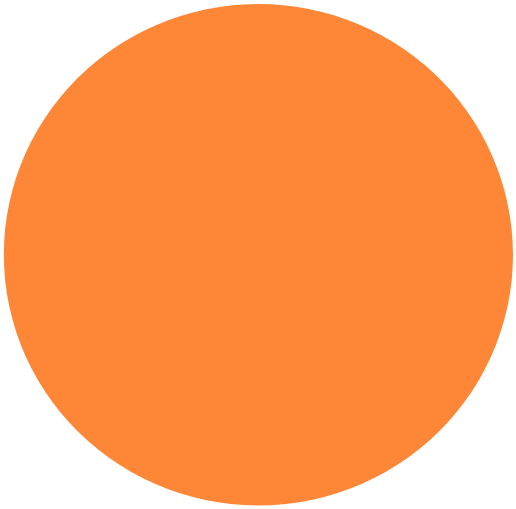
Χειμερινό Εξάμηνο 2020-2021

AMPHIPOLIS

Project 2020

Τζανάκης Νικόλαος
csd 4349
13/01/2020

Περιεχόμενα



1 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.

2 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 διαχείρισει του παίκτη και για το board.

Model.Tiles

Enum Colors

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

Abstract Class Tile

Με την 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 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

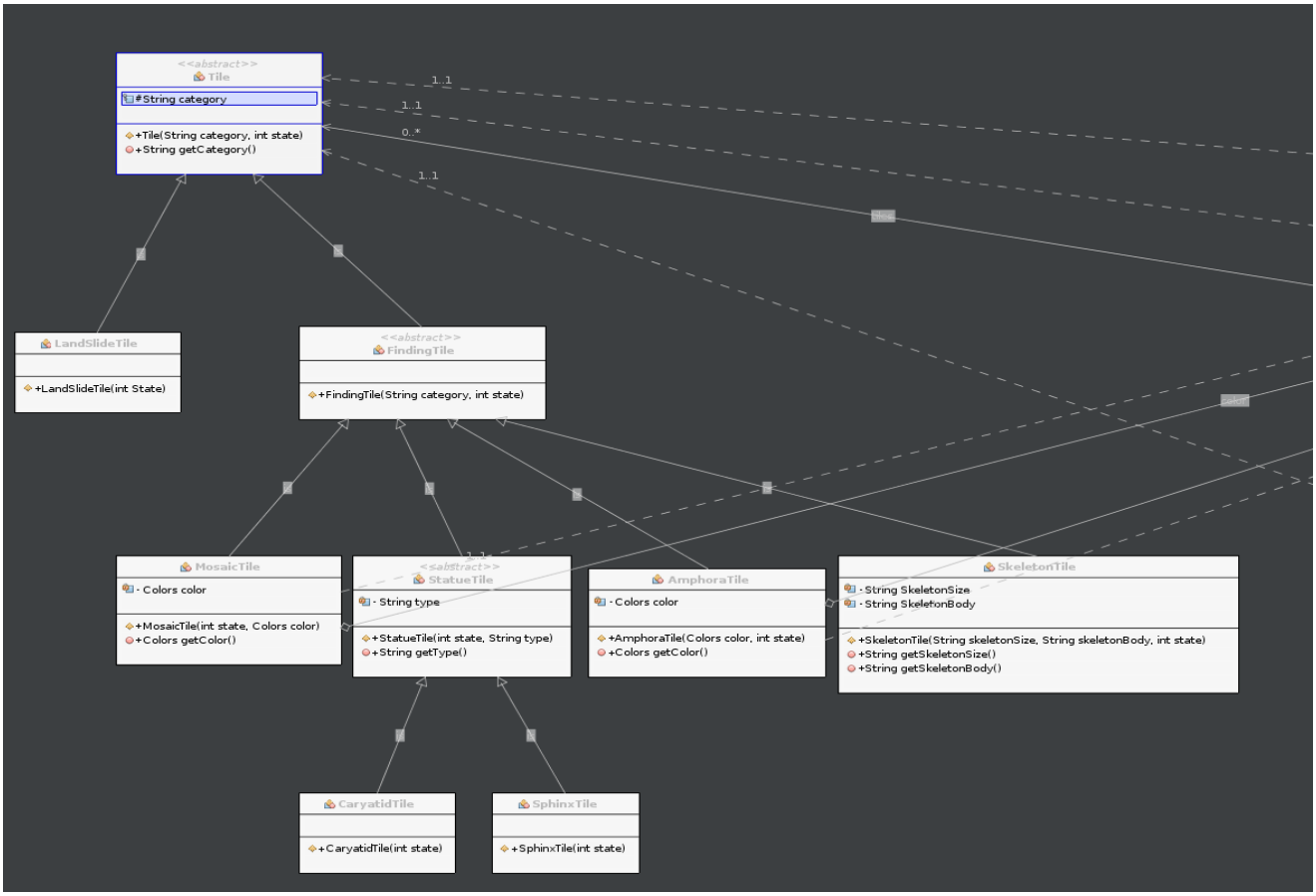
Class AmphoraTile, MosaicTile, SkeletonTile

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

Η 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

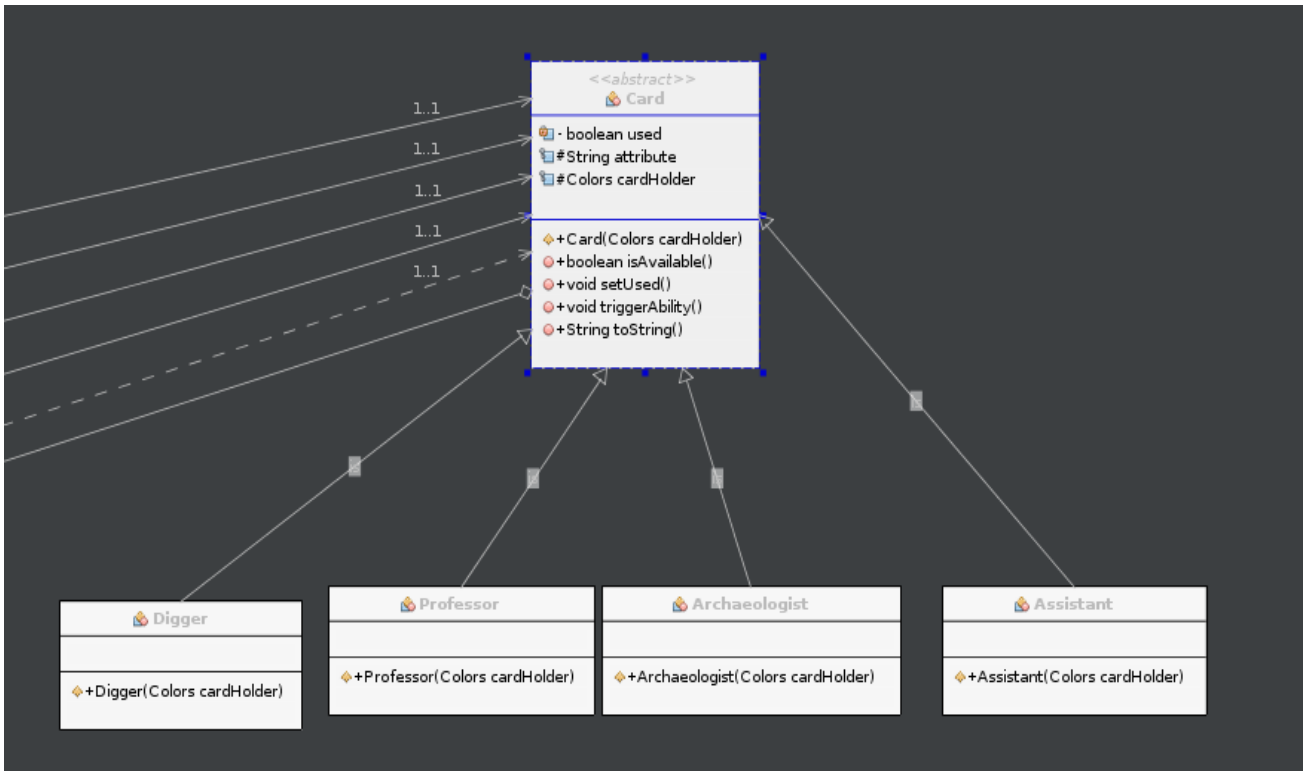
public void triggerAbility()

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

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 getMosaicPoints()

Υπολογίζει τους πόντους απο τα mosaics του παίκτη και τους επιστρέφει.

Public int getAmphoraPoints()

Υπολογιζει τους πόντους απο τα amphorae του παίκτη και τους επιστρέφει.

Public int getStatuePoints(int statuePoints[])

Επιστρέφει τους πόντους που αναλογούν στον παίκτη.

Επειδή τα statue points υπολογίζονται με βάση όλους τους παίκτες,υπολογίζονται στο Controller και περνάνε ως παράμετρο σε αυτη τη συνάρτηση για κάθε παίκτη,και η συνάρτηση επιστρέφει τους πόντους στον αντίστοιχο παίκτη

public int getSkeletonPoints()

Υπολογίζει τα skeletonPoints του παίκτη και τα επιστρέφει

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

public Tile drawTile(Collection bag)

Διαλέγει ένα τυχαίο tile απο το collection bag και το επιστρέφει.

Επισης αφαιρεί το tile που επιλέχτηκε απο το collection bag

Χρησιμοποιείται για να βάζουμε tiles στο board

public void getTile(Collection col)

Επιλέγει ένα τυχαίο tile απο το collection col και το προσθέτει στο collection με tiles του παίκτη.

Public Collection getPlayerCollection()

Επιστρέφει την collection με tiles του παίκτη

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.

Boolean checkCol(Tile I)

Επιστρέφει αν το collection κατηγορίας Tile ίδια με του I είναι άδειο

public Collection getAmphoraeCollection()

επιστρέφει το collection με τα amphorae Tiles

public Collection getMosaicCollection()

επιστρέφει το collection με τα mosaic Tiles

public Collection getLandSlideTilesCollection()

επιστρέφει το collection με τα landSlide Tiles

public Collection getSkeletonCollection()

επιστρέφει το Collection με τα skeleton Tiles

public Collection getStatueCollection()

επιστρέφει το Collection με τα statue Tiles

3 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)

Επίσης έχουμε κάποιες μεταβλητές private int hasPlayerState,playerThatPlays,pickTileState έτσι ώστε να οργανώνεται το round ενός παίκτη,αν μπορεί να τραβήξει tiles,απο ποια περιοχή και να γνωρίζουμε ανα πάσα στιγμή ποιός παίκτης παίζει εκείνο το round

Στον constructor της κλάσης Controller δημιουργούνται 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 int getPlayerState()

Επιστρέφει το state του player.

0 αν δεν έχει κάνει κίνηση.

1 μετά που τράβηξε tiles απο την σακούλα

2 μετά που πήρε 2 tiles απο το board.

3 αν χρησιμοποίησε κάρτα χαρακτήρα

public void setPlayerState(int hasPlayed)

κάνει το hasPlayedState ίσο με hasPlayed

Public Colors getCurrentPlayer()

Επιστρέφει τον παίκτη που παίζει στον γύρο που παίζεται εκείνη την ώρα ανάλογα με το χρώμα

public void setCurrentPlayer(Colors playerColor)

αναθέτει την μεταβλητη Colors playerThatPlays ως Colors playerColor

για να αλλάζουμε τον παίκτη που παίζει στο round μετα το endTurn

public Collection getBag()

επιστρέφει την σακούλα με τα Tiles του Controller

public Board getBoard()

επιστρέφει το board του Controller

public Player getPlayer1(),getPlayer2(),getPlayer3(),getPlayer4()

επιστρέφει τους παίκτες του παιχνιδιού ανάλογα ποιά μέθοδο καλέστηκε

public String getURL(Tile category)

επιστρέφει το URL της φωτογραφίας ανάλογα με την κατηγορία του Tile category

public int getPickTileState()

επιστρέφει την pickTileState μεταβλητή

public void setPickTileState(int i)

αναθέτει την τιμή του pickTileState ίση με i

public int[] calculateSphinxTilesForPlayers()

επιστρέφει έναν πίνακα με τον αριθμό των sphinx Tiles του κάθε παίκτη.

public int[] calculateCaryatidTilesForPlayers()

επιστρέφει έναν πίνακα με τον αριθμό των caryatid Tiles του κάθε παίκτη.

Public int[] calculateStatuePointsForPlayers()

υπολογίζει και επιστρέφει εναν πινακα με τα points που ανήκουν σε κάθε παίκτη ανάλογα με των αριθμό των Statue Tiles που έχουν οι παίκτες.

Όταν καποιοι παίκτες έχουν το ίδιο max αριθμο απο μιας κατηγορίας statue παίρνουν

6 πόντους ο καθένας

4 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 .

Το παιχνίδι ξεκινάει αμέσως μετά την εκτέλεση του προγράμματος.Έχουμε ενα Jpanel που πάνω σε αυτό διαδραματίζεται το παιχνίδι.

Αρχικά πάνω αριστερά,εμφανίζεται το menu το οποίο σου δίνει δύο επιλογές,να αρχίσει το παιχνίδι απο την αρχή,η να το κλείσεις.

Μεσα στο panel εχουμε τις περιοχές των tiles οι οποίες ανοίγουν με το ανάλογο button.

Επίσης πάνω δεξιά υπάρχουν 3 Jlabel ,όπου το πρώτο δείχνει το όνομα του παίκτη που παίζει στο συγκεκριμμένο round,το 2ο δείχνει το state που βρίσκεται το round,δηλαδη αν ο παίκτης πρέπει να τραβήξει tiles απο την σακούλα,αν πρέπει να πάρει tiles απο το board,και αν τελείωσε ο γύρος του.Το 3ο γράφει Use character για τους χαρακτήρες.

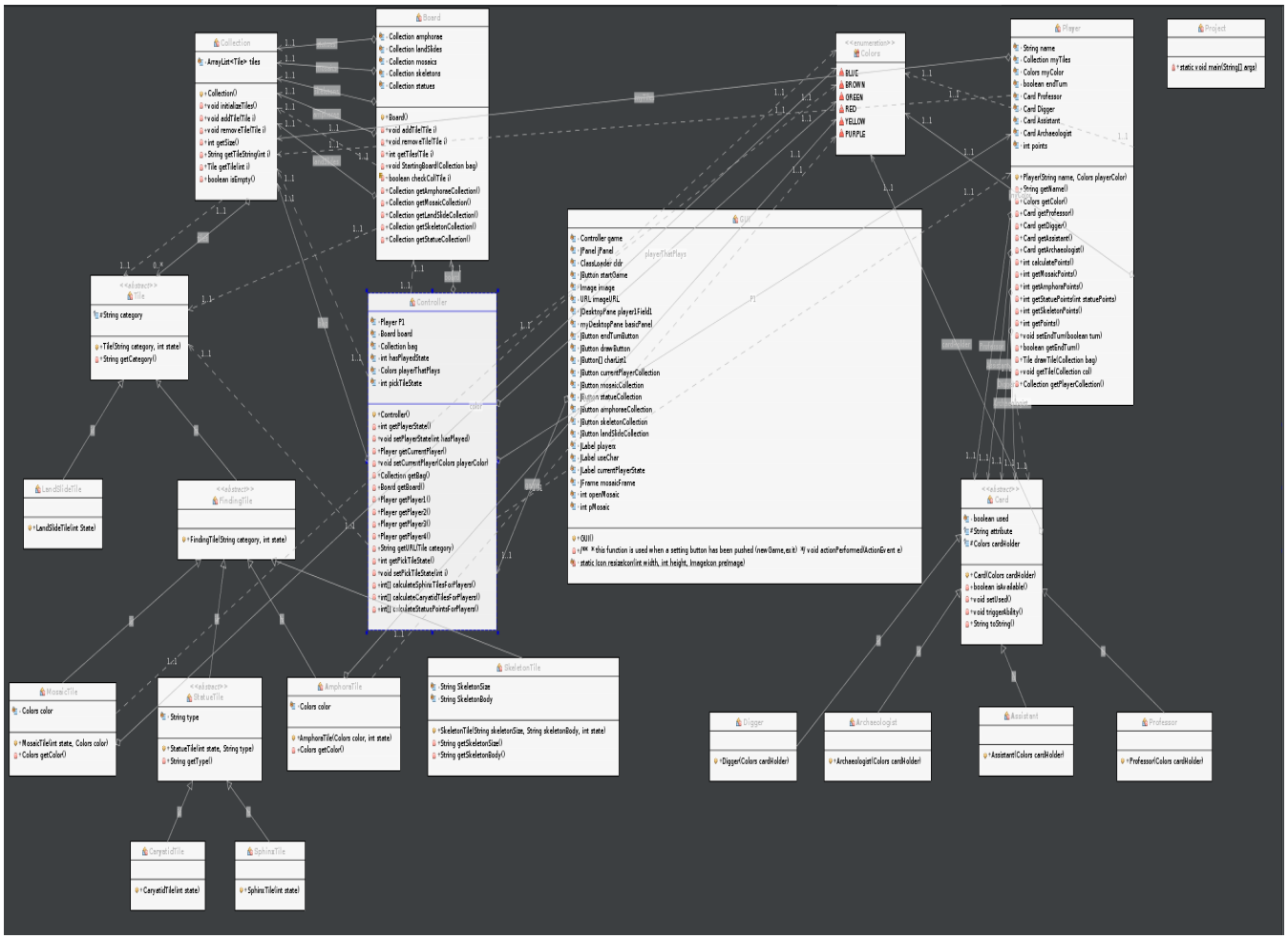
Ακόμα στο panel στη μέση δεξιά υπάρχουν γραφικά οι κάρτες χαρακτήρα (όπου δυστυχώς δεν τις υλοποίησα),και ακριβως απο κάτω 2 buttons ,οπου με το Draw Tiles τραβάμε κάρτες απο την σακούλα,και το End Turn τερματίζει τον γύρο του παίκτη,και πάει στον επόμενο παίκτη.Τέλος στη κάτω μεριά,υπάρχουν 4 JButtons τα οποία χρησιμοποιούνται για να δούμε τα collections των players σε κάθε δεδομένη στιγμή,αφου τα tiles των παικτών πρέπει να είναι ορατά σε όλους.

Ακόμα στην κλάση GUI υπάρχουν τα ActionListeners των Jbuttons που υπάρχουν στο Jpanel.

Αυτό που δεν ανέφερα παραπάνω ειναι τα Jbuttons των tiles του board,τα οποία όταν έχουμε την δυνατότητα να επιλέξουμε κάποιο tile,μόλις το επιλέξουμε το παράθυρο κλείνει και ξαναανοίγει(ξαναπατιέται το κουμπι του collection που πήραμε τα tiles)

Αν δεν έχουμε την δυνατότητα να επιλέξουμε κάποιο tile,απλα πατιέται το κουμπι χωρις να γίνεται τίποτα,και αν εχουμε την δυνατότητα να επιλέξουμε 2ο tile αλλα έχουμε ανοίξει διαφορετικό collection για να το πάρουμε απο αυτό του 1oy tile που επιλέξαμε,κλείνει και ξαναΑνοίγει το παράθυρο

5 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 τον controller,οπου έχει χαρακτηριστικά απο την κλάση player,την κλάση board την κλάση Collection και την κλάση Colors

Απο την δεξιά μεριά έχουμε 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(Project) οπου απλα θα καλεί το GUI

6 Λειτουργικότητα (Β Φάση)

Γενικά νομίζω ότι υλοποίησα ολα τα ερωτήματα επιτυχώς,εκτος τα Junit Tests,με τα οποία δεν ασχολήθηκα καθόλου,και την λειτουργία των καρτών χαρακτήρα.Στίς κάρτες χαρακτήρα με δυσκόλεψε το γραφικό κομμάτι,διότι είχα φτιάξει διαφορετικά jPanel για κάθε παίκτη, και είχα προσθέσει τις εικόνες characterButtons[] αλλα όταν πρόσθετα ενα characterButton σε ενα Jpanel εκτος του 1ου παίκτη,εξαφανιζόταν και απο τον πρώτο παίκτη με αποτέλεσμα να μην φαινόταν καθόλου η εικόνα στα γραφικά.Βέβαια,δεν ασχολήθηκα και πολυ ,όπως δεν ασχολήθηκα καθόλου με τα Junit tests διότι δεν έχω καμιά εμπειρία και με πίεζε ο χρόνος

<u>ΑΛΛΑΓΕΣ ΣΕ ΣΧΕΣΗ ΜΕ ΤΗΝ ΣΧΕΔΙΑΣΗ ΤΗΣ Α' ΦΑΣΗΣ</u>

Παραδόξως,δεν έγιναν πολλές αλλαγές με τη σχεδίαση της Α φάσης.Αυτο που άλλαξε,ειναι φυσικά ο κώδικας,και σε καποιες κλάσεις όπως το Controller,προστέθηκαν κάποιοι μεθόδοι για την λύση κάποιων προβλημάτων.Η σχεδίαση της Α φάσης με βοήθηκε αρκετά και με καθοδήγησε στην συγγραφη του κώδικα.Επίσης,αρχικά θα έκανα ένα μενού,αλλά το αφέρεσα στη Β φάση.Επιπλέον ,έγιναν και κάποιες αλλαγές για το πως θα επιλέγουμε ποιος παίκτης παίζει σε ενα round.

<u>ΧΕΙΡΙΣΜΟΣ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ</u>

Ο χειρισμός του παιχνιδιού νομίζω είναι αρκετά εύκολος,βοηθάει και το συγκεκριμμένο παιχνίδι σε αυτό βέβαια.

Προγραμματιστικά,δυστυχώς βιάστικα σε κάποια πράγματα που ίσως θα ήθελα να είναι διαφορετικά,όπως να απεύφευγα κάποια IF statements ,και να έκανα πιο πολλούς μεθόδους για κάποιες λειτουργίες που έγιναν μέσα στο GUI.

7 Συμπεράσματα

Γενικά,ήταν μια πολύ καλή εμπειρία και ένα δυνατό project για να ασχοληθούμε με τις κλάσεις και τις λειτουργίες τους,την java,όπως επίσης και με ενα μέρος των γραφικών της Java.

Η σχεδίαση της Α φάσης ήταν καθοριστική για την υλοποίηση της εφαρμογής,βοήθησε πολύ και γενικά υπήρχε μια καθοδήγηση στον κώδικα.Επίσης το μοντέλο MVC (Model View Controller) φάνηκε να ειναι αρκετά αξιόπιστο και εύκολο στην διαχείριση και κατανόηση του.Φυσικά,θα ήθελα να αλλάξω πολλά πράγματα και να κάνω τον κώδικα πιο μαζεμένο και καλύτερο,όπως επίσης και κάποιες σημειώσεις πάνω στα γραφικά(π.χ να γράφει λεπτομέρειες για το state του παιχνιδιού)

Γενικά δεν αντιμετώπισα κάποιο πρόβλημα,μόνο ίσως με πέδεψαν τα Γραφικά λίγο αφού δεν είχα και κάποια προηγούμενη εμπειρία.