

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2020-2021**

Project Φάση B

**

Θεόδωρος Χαλκίδης

csd4198

2/12/2020

**Περιεχόμενα**

[**1.**](#_gjdgxs) **Εισαγωγή 1**

[**2.**](#_30j0zll) **Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 1**

[**3.**](#_1fob9te) **Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller** 6

[**4.**](#_3znysh7) **Η Σχεδίαση και οι Κλάσεις του Πακέτου View** 7

[**5.**](#_2et92p0) **Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML** 8

[**6.**](#_tyjcwt) **Λειτουργικότητα (Β Φάση)** -

[**7.**](#_3dy6vkm) **Συμπεράσματα** -

## Εισαγωγή

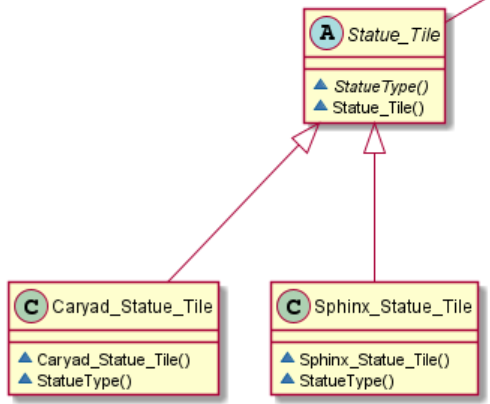
Το πρότζεκτ είναι χωρισμένο σύμφωνα με το μοντέλο MVC (Model / View / Controller ). Το model κομμάτι περιέχει όλες τις κλάσεις tiles, board , players, characters etc οι οποίες αποτελούν τον πυρήνα του παιχνιδιού καθώς περιγράφουν τα βασικά στοιχεία του επιτραπέζιου. Το view κομμάτι περιέχει των τρόπο με τον οποίο θα στηθούν τα γραφικά , δηλαδή την σχεδίαση του gui . Τέλος το controller κομμάτι περιέχει τις εκτελεστικές μεθόδους που χρειάζονται ώστε να ενώσουν το model με το view καθώς και ελέγχει αν το παιχνίδι τελείωσε . Επίσης το controller είναι υπεύθυνο για την λειτουργία save / load και για την αρχικοποίηση των λειτουργιών των γραφικών

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

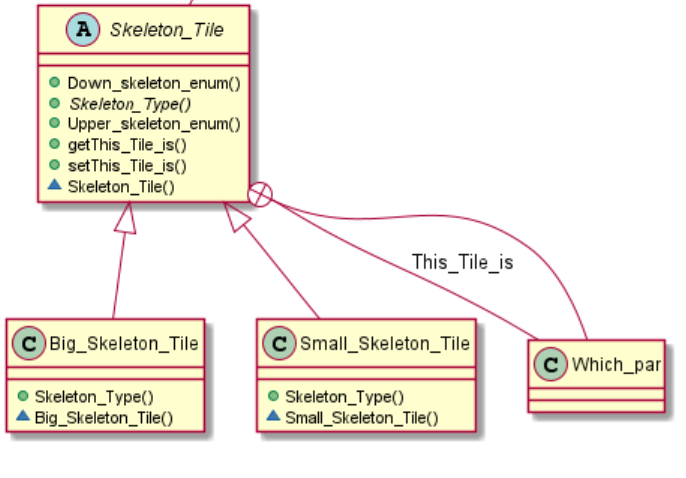
## Κλάση Tile και υποκλάσεις της :

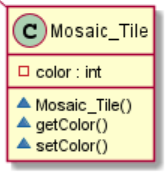
**Η κλάση Tile** έχει 3 μεθόδους , μία από αυτές είναι κατασκευάστρια, και ένα string το οποίο αποθηκεύει την εικόνα του tile. Αυτές οι μέθοδοι καθώς και το string είναι κοινά για όλων των ειδών τα tiles και για αυτό επέλεξα να τα έχω στο super class Tile . Οι δύο μέθοδοι που έχω θέσει είναι μία set και μία get για το string . Σε όσες κλάσεις δεν αναφέρεται κατασκευάστρια μέθοδος εννοείτε της Tile.

**Η κλάση Landslide\_Tile** είναι άμεσο παιδί της Tile και είναι τα tile που μπαίνουν στην είσοδο , που όταν γεμίσει τελειώνει το παιχνίδι. Δεν φέρουν κάποιο διαφορετικό χαρακτηριστικό από τα βασικά του Tile και αυτό καθώς πιο πολύ λειτουργούν ως ένας αντίστροφος μετρητής.

**Η κλάση Finding\_Tile** είναι child της Tile και είναι mother όλων των κλάσεων/tile(4) με τα οποία θα αλληλεπιδρά ο παίκτης.

**Η κλάση Statue\_Tile** είναι child της Tile και είναι mother των κλάσεων Caryad\_Statue\_Tile και Sphinx\_Statue\_Tile . Έχει μία abstract μέθοδο , που υλοποιείται στις κλάσεις παιδιά , για να επιστρέφει με string το τι είδους είναι το παίδι. Τα παιδιά της πέρα από την υλοποίηση του StatueType δεν φέρουν κάποιο άλλο γνώρισμα.

**Η κλάση Skeleton\_Tile** είναι child της Tile και είναι mother σε δύο κλάσεις παιδιά την Small\_Skeleton\_Tile και την Big\_Skeleton\_Tile . Η κλάση έχει εναν enum που αποθηκεύει άμα το skeleton tile είναι άνω κομμάτι ή κάτω κομμάτι , καθώς έχει και τις κατάλληλες μεθόδους ώστε να θέτει και να βλέπει τι είδους κομμάτι είναι αλλά και σε τι είδους skeleton tile (big || small) ανήκει . Τα δύο παιδιά της κλάσης έχουν υλοποίησει την μέθοδο που επιστρέφει τι είδους skeleton tile (big || small) είναι και δεν φέρουν κανένα άλλο χαρακτηριστικό. Το χαρακτηριστικό This\_Tile\_Is αποθηκεύει το enum Which\_par. Ακόμα η κατασκευάστρια μέθοδος θα πρέπει να ορίζει και το enum.

**Η κλάση Amphora\_Tile** είναι child της Tile και έχει 1 εξτρά χαρακτηριστικό σε σχέση με την Tile το οποίο είναι το χρώμα του amphora tile . Είναι αποθηκευμένο σαν int με τις τιμες 1,2,3,4,5,6 να παίρνουν αντίστοιχα τα χρώματα μπλε, καφέ, κόκκινο, πράσινο, κίτρινο και μωβ. Επίσης έχει μία set και μία get μέθοδο οι οποίες αφορούν το χρώμα.Ακόμα η κατασκευάστρια μέθοδος θα πρέπει να αναθέτει και το color .

**Η κλάση Mosaic\_Tile** είναι child της Tile και έχει 1 εξτρά χαρακτηριστικό σε σχέση με την Tile το οποίο είναι το χρώμα του amphora tile . Είναι αποθηκευμένο σαν int με τις τιμες 1,2,3 να παίρνουν αντίστοιχα τα χρώματα πράσινο, κίτρινο και κόκκινο. Επίσης έχει μία set και μία get μέθοδο οι οποίες αφορούν το χρώμα. Ακόμα η κατασκευάστρια μέθοδος θα πρέπει να αναθέτει και το color .

## Κλάση Character και υποκλάσεις της :

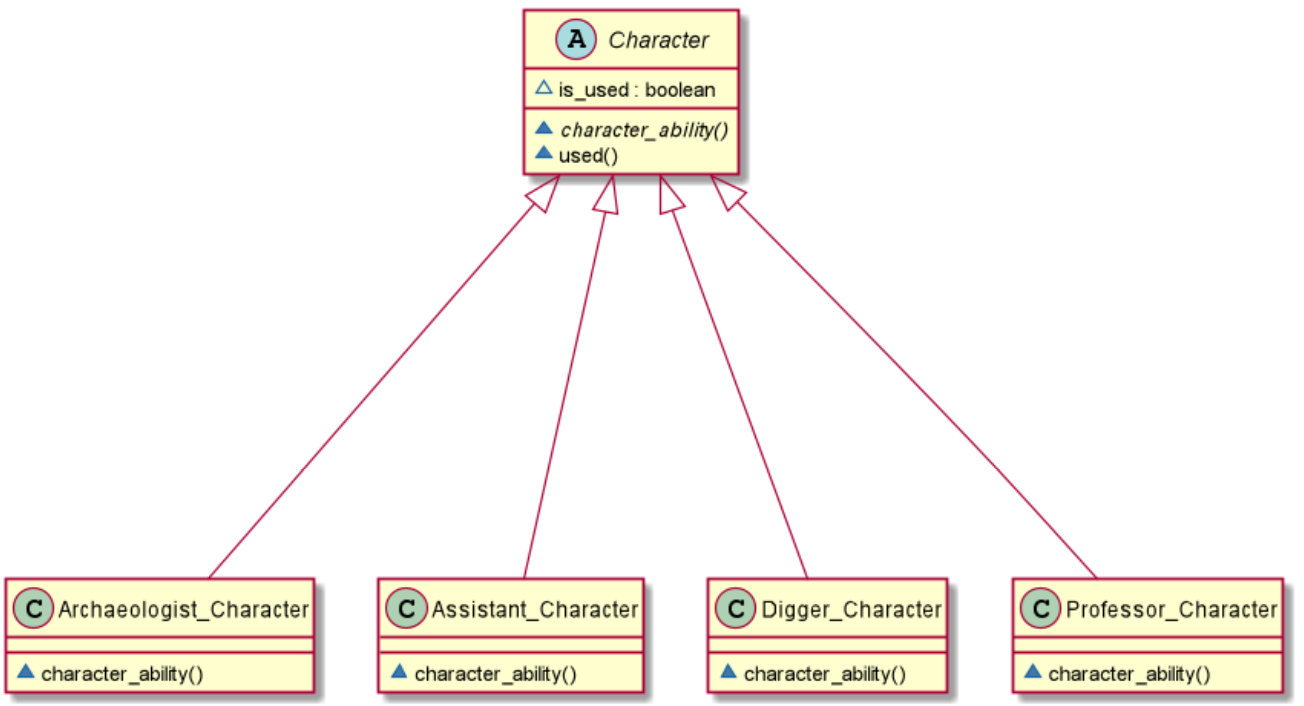
**Η abstract κλάση Character** είναι η mother κλάση (4) κλάσεων . Περιέχει μία μεταβλητή τύπου boolean η οποία αποθηκεύει εάν ο χαρακτήρας χρησιμοποίησε την abstract μέθοδο ability που την θα την υλοποιήσουν καθένα από τα 4 παιδιά της ξεχωριστά. Ακόμα περιέχει την μέθοδο used που είναι κοινή για όλα τα παιδιά κλάσεις και που αλλάζει την boolean σε TRUE άμα χρησιμοποιήθηκε η μέθοδος ability .

Καθένα από τα 4 παιδιά της Character έχει μία διαφορετική ability μέθοδο η οποία μπορεί να χρησιμοποιηθεί μόνο μία φορά .

Archaeologist : Ο παίκτης παίρνει μέχρι 2 πλακίδια από οποιαδήποτε περιοχή τοποθέτησης, εκτός από αυτή που διάλεξε νωρίτερα στη σειρά του.

Assistant : Ο παίκτης παίρνει 1 πλακίδιο από οποιαδήποτε περιοχή τοποθέτησης.

Digger : Ο παίκτης παίρνει μέχρι 2 πλακίδια από την περιοχή που διάλεξε νωρίτερα στη σειρά του.

Professor : Ο παίκτης παίρνει ένα πλακίδιο από κάθε περιοχή, εκτός από αυτή που διάλεξε νωρίτερα στη σειρά του.

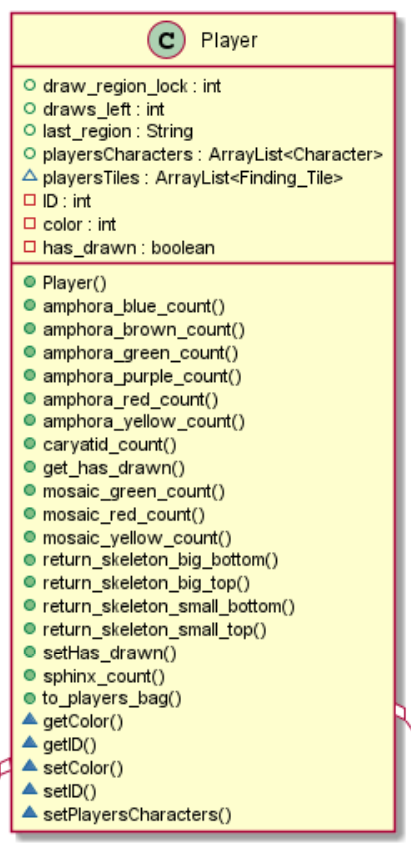
## Κλάση Bag :

Η κλάση bag είναι υπεύθυνη για την αρχικοποίηση και την αποθήκευση των tiles πριν μπουν στο ταμπλό , καθώς και για το τυχαίο τράβηγμα τους. Έχει ένα arraylist από tiles και μία rand η οποία σαν seed παίρνει τον χρόνο για την βέλτιστη επίτευξη τυχαιότητας. Έχει μία κατασκεύαστρια κλάση η οποία καλεί την initialise\_bag η οποία αρχικοποιεί την bag και βάζει τα tiles μέσα στο array list. Ακόμα έχει τις μεθόδους gettilefrombag η οποία τραβάει τυχαία ενα tile από το array list , την how\_many\_left η οποία επιστρέφει πόσα tiles έχουν μείνει , έχει την tobag η οποία βάζει πίσω tile.

## Κλάση Player:

Η κλάση player είναι υπεύθυνη για την αποθήκευση πληροφοριών του παίκτη . Έχει ένα arraylist από characters που αποθηκεύει τους 4 characters του χρήστη . Έχει ένα array list από finding tiles για να αποθηκεύει τα tiles που ο παίκτης έχει πάρει . Έχει και 2 χαρακτηριστίκα για να τον καθορίζουν , το ID και το Color (int 1,2,3,4 == κίτρινο, κόκκινο, μπλε, μαύρο). Ακόμα υλοποιεί τις μεθόδους totalpoint που υπολογίζει τους πόντους που έχει μαζέψει ο παίκτης, τις μεθόδους set και get για τα id και color καθώς και την setplayerscharacters που αρχικοποιεί τους χαρακτήρες για κάθε παίκτη. Τέλος έχει και την κατασκευάστρια μέθοδο του που θα θέτει το id , color και θα αρχικοποίει τα array list του κάθε παίτη . Για την Β φάση έβαλα συναρτήσεις που επιστρέφουν πόσα από το κάθε Finding\_Tile έχει ο κάθε χρήστης. Επίσης έβαλα το boolean has\_drawn που αποθηκεύει αν ο χρήστης έχει τραβήξει αυτό τον γύρο. Ακόμα έβαλα την to player\_bag που βάζει στου παίκτη το πουγκί αυτά που επέλεξε.

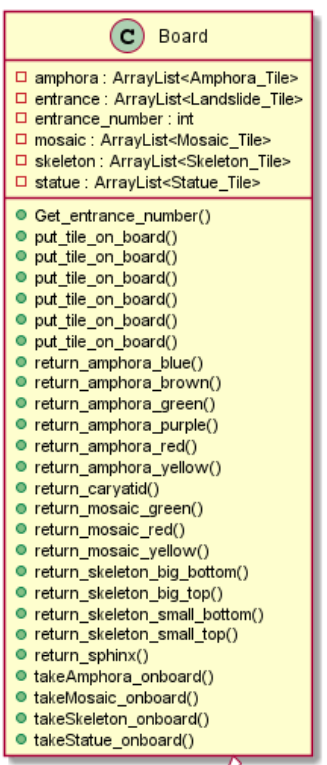
**ΑΝΑΝΕΩΜΕΝΟ ΓΙΑ Β ΦΑΣΗ PLAYER UML**

****

## Κλάση Board:

Η κλάση board είναι υπεύθυνη για ότι υπάρχει στο ταμπλό . Έχει 5 array list ένα για κάθε είδος από tile καθώς και έναν αριθμό που κρατάει πόσα landslide tile έχουν μπει έτσι ώστε να ξέρουν οι παίκτες πόσο έμεινε στο παιχνίδι . Υλοποιεί τις μεθόδους put\_tile\_on\_board οι οποίες βάζουν το tile στο κατάλληλο array list , και 4 μεθόδους take\_X\_onboard (οπου X το κατάλληλο tile) οι οποίες διαγράφουν από το board το tile και το επιστρέφουν ώστε να προστεθεί στον κατάλληλο player. Επίσης υπάρχει μία get\_entrance\_number η οποία επιστρέφει πόσα landslide tiles υπάρχουν στο ταμπλό. Για την Β φάση πρόσθεσα μεθόδους που επιστρέφουν πόσα από το κάθε είδος Tile υπάρχουν στο ταμπλό

**ΑΝΑΝΕΩΜΕΝΟ ΓΙΑ Β ΦΑΣΗ BOARD UML**

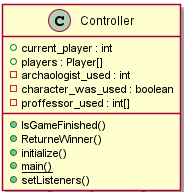


## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

To Controller είναι υπεύθυνο για την σωστή διεκπεραίωση του παιχνιδιού καθώς και για το πάντρεμα view model . Θα πρέπει να έχει τις μεθόδους isgamefinished που ελέγχει αν το παιχνίδι τελείωσε , την μέθοδο nextturn για να γίνονται refresh τα gui για τον επόμενο γύρο , την μέθοδο after\_first\_pick που αφού ο παίκτης διαλέξει ένα είδος tile θα πρέπει να εξασφαλίζει ότι θα πάρει από το ίδιο, θα έχει την μέθοδο initialize όπου θα αρχικοποιούνται ότι χρειάζεται το παιχνίδι για να τρεξει , θα έχει την μέθοδο set\_listeners που θα αντιστοιχεί τα κουμπιά του gui σε διάφορες λειτουργίες ,θα έχει τις μεθόδους savegame και loadgame για να αποθηκεύει και να φορτώνει το παιχνίδι και θα έχει την κατασκευάστρια μέθοδο του που θα καλει το initialize και το set\_listeners και θα βλέπει άμα μπορεί να κάνει load .

Για την Β φάση τελικά δεν χρειάστηκα τα nextturn , after\_first\_pick , load\_game , save\_game . Αντιθέτως δημιούργησα τα archaologist\_used που κρατάει άμα ο αρχαιολόγος έχει χρησιμοποιηθεί αλλά και το που , character\_used που κρατάει αμα παίχτηκε κάρτα χαρακτήρα αυτό τον γύρο, professor\_used που κρατάει άμα ο proffessor έχει χρησιμοποιηθεί αλλά και το που. Ακόμα πρόσθεσα την returnwinner η οποία στο τέλος μετράει τους πόντους και επιστρέφει τον νικητή

**ΑΝΑΝΕΩΜΕΝΟ ΓΙΑ Β ΦΑΣΗ CONTROLLER UML**

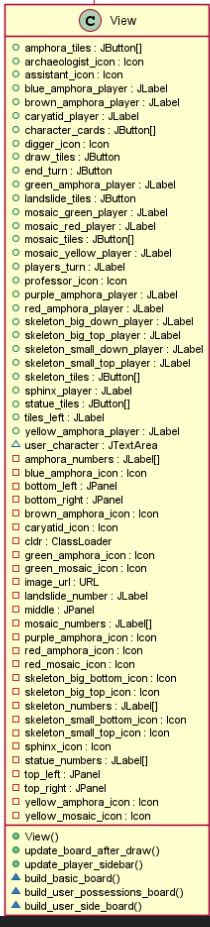


## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

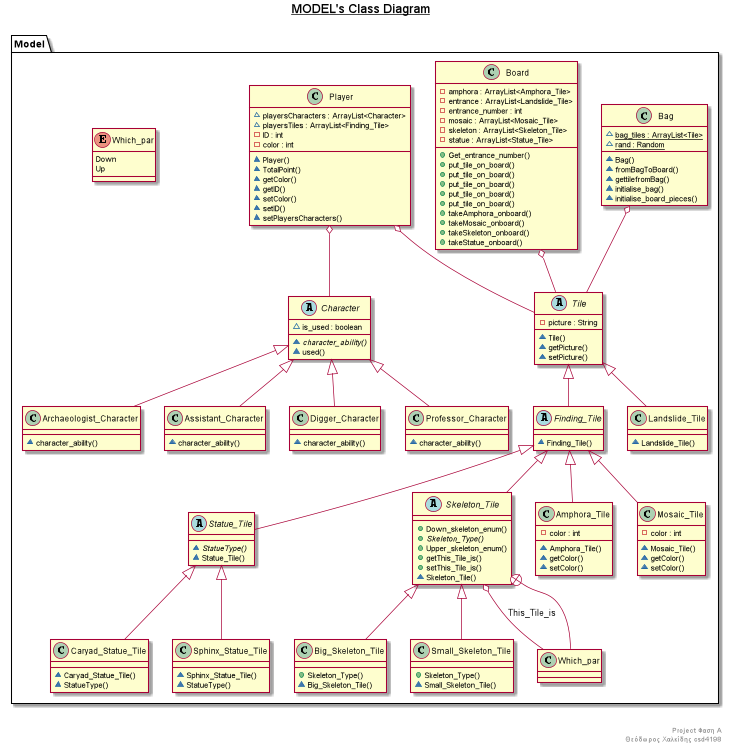
Το view είναι υπεύθυνο για τα γραφικά του παιχνιδιού. Έχει ένα σετ κουμπιών για κάθε είδος tile , ένα κουμπί draw tiles , ένα κουμπί next turn , 4 κουμπιά χαρακτήρα καθώς και δύο περιοχές κειμένου (μία για το ποια σειρά είναι και μία με ένα μικρό κειμενάκι). Αποτελείται από τις μεθόδους build\_basic\_board , build\_user\_possessions\_board , build \_user\_side\_board που δημιουργούν τα 3 ταμπλό του παιχνιδιού καθώς και απο τις μεθόδους υπεύθυνες για τα update αυτών με ονόματα update\_board\_after\_draw για update μέτα το τράβηγμα από το bag, update\_board\_after\_end\_turn για update μετά το τέλος γύρου, updated\_board\_after\_pick για update μετά το τράβηγμα tile από το ταμπλό και το update\_player\_after\_use\_character για το update μετά που θα χρησιμοποιηθεί ένας χαρακτήρας . Όλα τα σετ κουμπιών δεν θα ξεπερνούν τα διαφορετικά είδη πλακιδίων πχ αν έχω 5 sphinx θα δείχνει 1 κουμπί με τον αριθμό 5 πάνω από τα γραφικά για εξοικονόμηση χώρου .

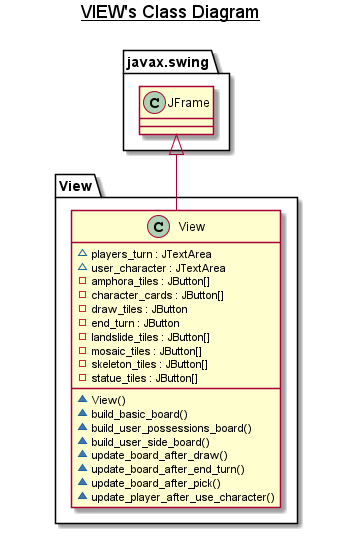
Για την Β φάση δημιούργησα ενα κουμπί και ένα jlabel για κάθε στοιχείο που θα υπάρχει στα γραφικά και τα οποία θα είναι public ώστε ο controller να μπορεί να τα ενώσει με το model . Ακόμα δεν χρειάστηκα τις update\_board\_after\_end\_turn, update board after pick , update board after use character και δημιούργησα την update player sidebar η οποία ανανεώνεται μετά από κάθε πράξη του χρήστη.

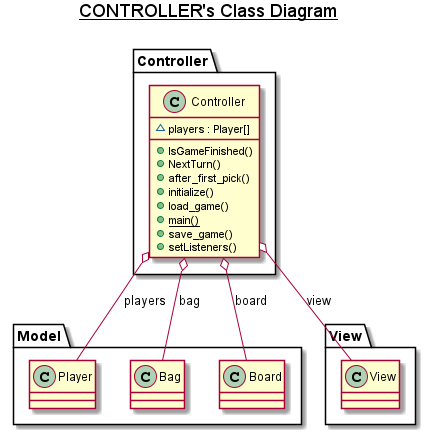
**ΑΝΑΝΕΩΜΕΝΟ ΓΙΑ Β ΦΑΣΗ VIEW UML**

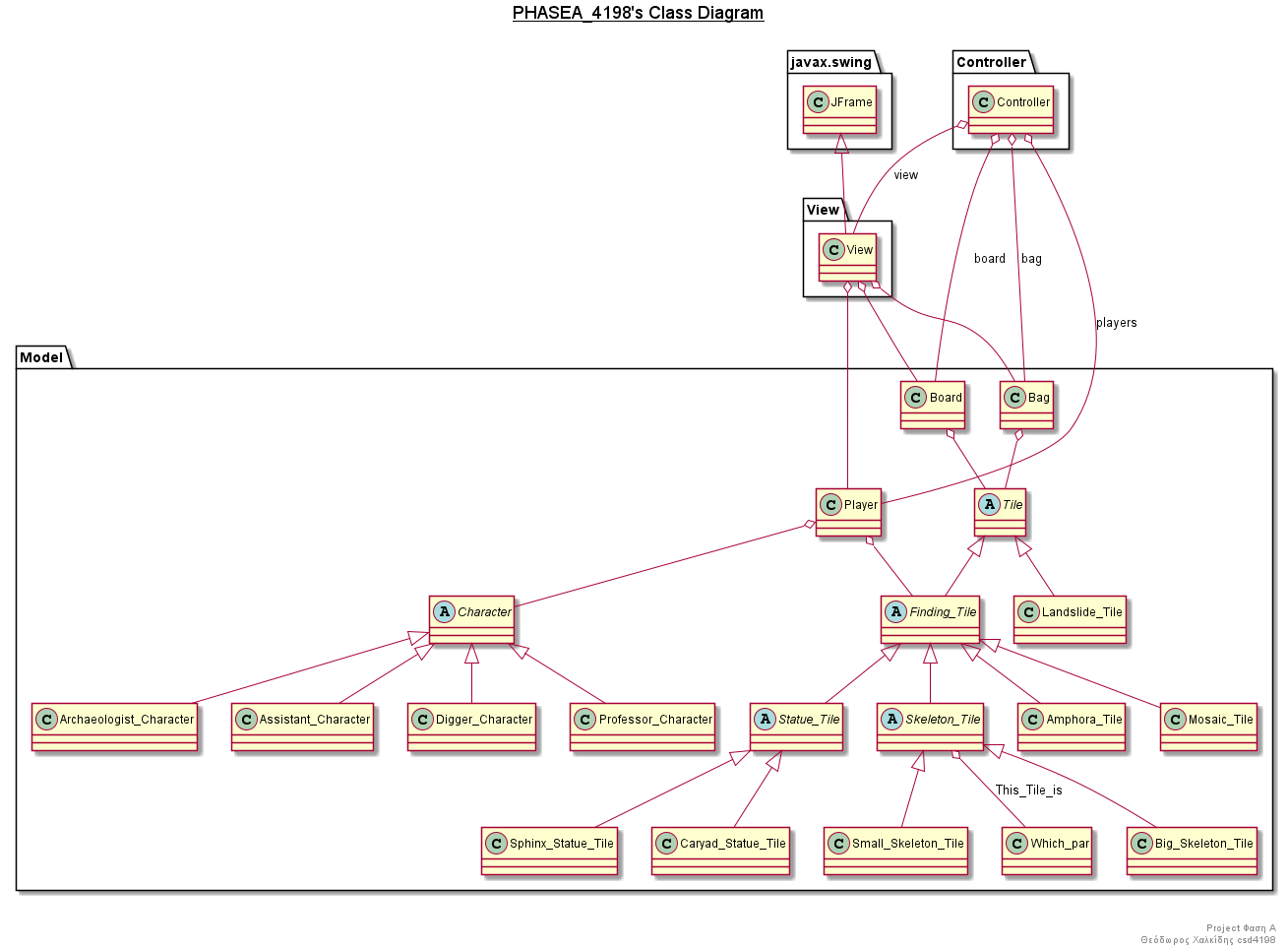
****

## Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

**Uml of Model**

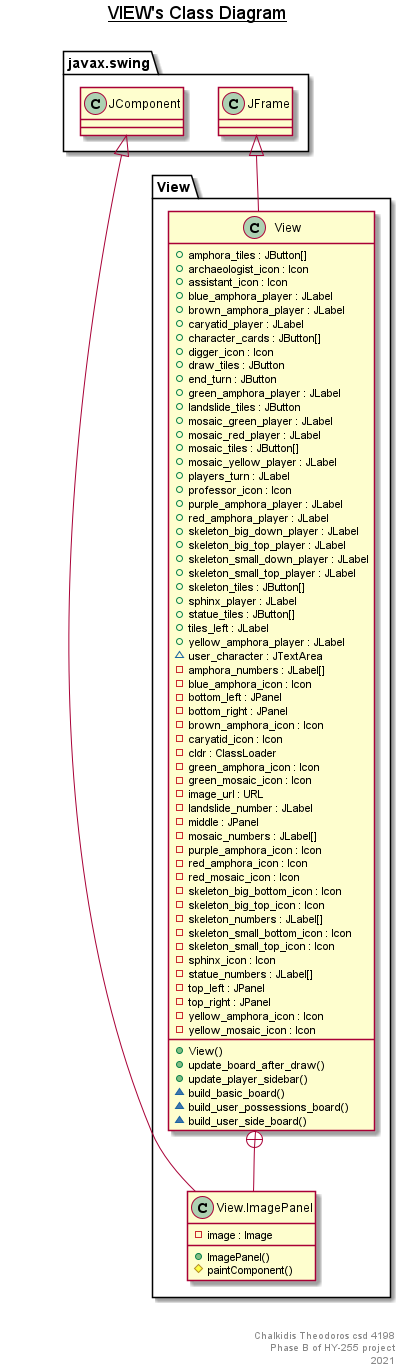
**Uml of View Α φάση**

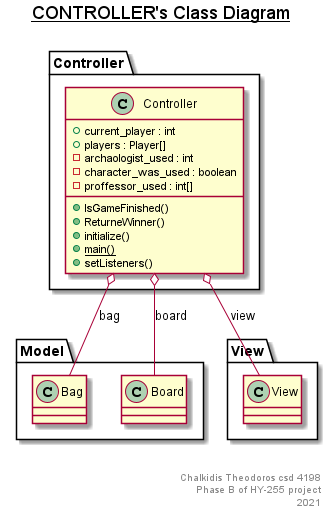
**Uml of Controller**

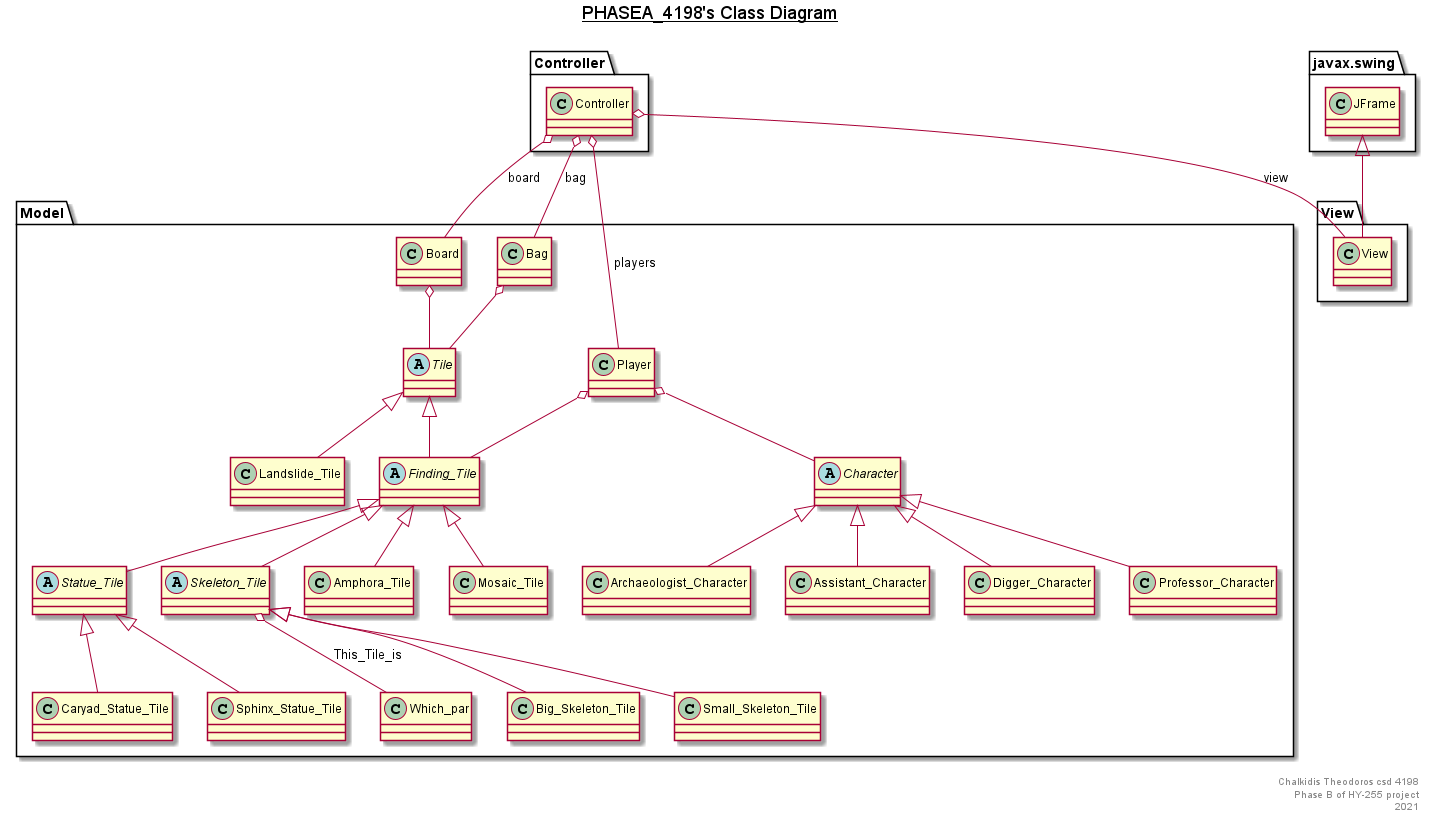
**Uml of the project phase A**

**UML Β φάσης**

(εκτός Model που έμεινε ίδιο)

****

****

****

## Λειτουργικότητα (Β Φάση)

Κατάφερα να ολοκληρώσω επιτυχώς όλα τα ερωτήματα της βασικής άσκησης . Απέφυγα να ασχοληθώ με τα bonus λόγω έλλειψης χρόνου.

## Συμπεράσματα

Κατάφερα να ολοκληρώσω επιτυχώς όλα τα ερωτήματα. Με δυσκόλεψαν ελαφρώς τα character cards καθώς και το πως θα δείχνω πόσα tile από το κάθε είδος υπάρχουν στο board . Το όλο project ήταν ενδιαφέρον απλά το επιλεγμένο επιτραπέζιο ήταν πολύ κακό και σίγουρα κάτι που δεν θα αγόραζα να παίξω .