



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

HY252– Αντικειμενοστραφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2024-2025

Αναζητώντας τα χαμένα Μινωικά Ανάκτορα

Μέρος Α΄



Σούμπασης Τραϊανός Αλέξανδρος

AM: 5207

15-01-2024

Περιεχόμενα

1. Εισαγωγή.....	3
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model.....	3
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller.....	11
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	12
5. Η αλληλεπίδραση μεταξύ κλάσεων – Διαγράμματα UML.....	13
6. JUnitTests.....	13
7. Συμπεράσματα	

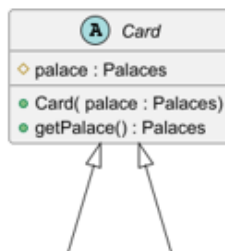
1. Εισαγωγή

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε το μοντέλο MVC, το οποίο χωρίζει την υλοποίηση της εργασίας σε τρία μέρη που υλοποιούνται ως πακέτα: Model, View, Controller. Το κάθε πακέτο περιέχει τις απαραίτητες δηλώσεις κλάσεων, οι οποίες με τη σειρά τους περιέχουν μεθόδους και μεταβλητές που θα αναλύσω στις παρακάτω ενότητες. Μερικές από αυτές είναι η αφηρημένη κλάση *Card* και η κλάση *Player* του πακέτου model, ή η κλάση *GraphicUI* του πακέτου View.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακέτο Model είναι με διαφορά το μεγαλύτερο πακέτο που περιέχει τις περισσότερες κλάσεις. Πιο αναλυτικά:

α. Η αφηρημένη κλάση *Card*:

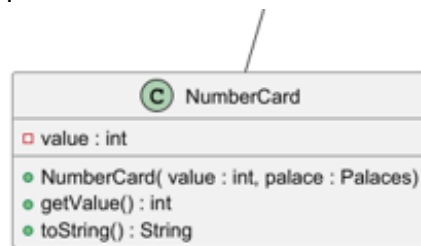


Η αφηρημένη κλάση *Card* περιέχει:

- Τη μεταβλητή *palace* τύπου *Palaces*(βλ.σελ.5)
- Τον constructor *Card* με παράμετρο *palace* τύπου *Palaces*
- Έναν getter *getPalace*
- Μια void συνάρτηση *applyEffect*
- Μια *toString*

Και λειτουργεί ως ένα “blueprint” για την σχεδίαση των διαφόρων τύπων καρτών που ακολουθούν.

i. Η κλάση NumberCard:



Η κλάση `NumberCard` κληρονομεί την `Card` και προσθέτει:

- Μια `int` τιμή `value`
- Έναν `constructor` που φτιάχνει μια `NumberCard` με τιμή `value` σε παλάτι `palace`.
- Έναν `getter` `getValue`.

Η `NumberCard` αφορά μια κάρτα που ο παίχτης χρησιμοποιεί για να μετακινήσει το πιόνι του σε ένα μονοπάτι κατά `value`.

ii. Η κλάση Special Card:



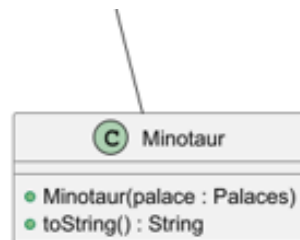
Η `Special Card` δεν υλοποιεί καμία μέθοδο παρά τον `constructor` γιατί υπάρχει για να την κληρονομήσουν οι επόμενες δύο.

1. Η κλάση Ariadne:



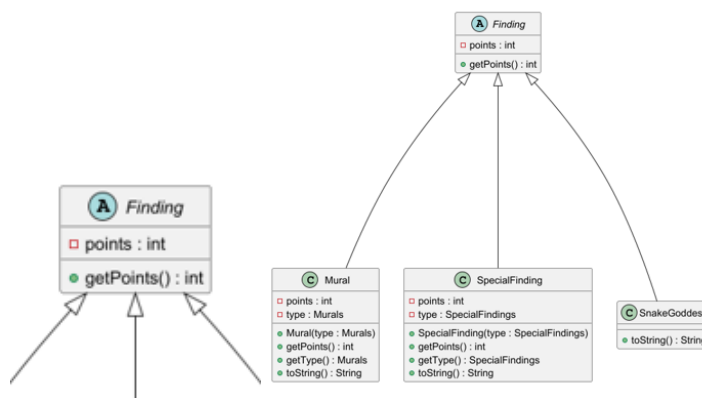
Η `Ariadne` περιέχει έναν `constructor` που την αρχικοποιεί με ένα `palace` και μια `toString` και αφορά μια κάρτα μίτου Αριάδνης που μετακινεί έναν παίχτη κατά δυο θέσεις.

2. Η κλάση Minotaur:



Η *Minotaur* αφορά μια κάρτα Μινώταυρου η οποία χρησιμοποιείται για να πάει το πιόνι ενός αντιπάλου πίσω κατά δύο θέσεις σε ένα συγκεκριμένο μονοπάτι.

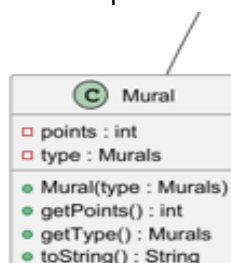
α. Η αφηρημένη κλάση *Finding*:



Η *Finding* λειτουργεί ως βάση για να φτιαχτούν οι επόμενες κλάσεις. Περιέχει ένα πεδίο με τους πόντους που δίνει το εύρημα και έναν getter για αυτό.

2.2.1

Η κλάση *Mural*:

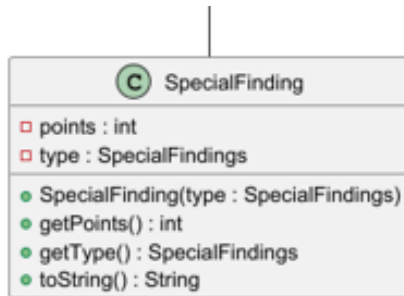


Η *Mural* κάνει extend την *Finding* και αφορά τις τοιχογραφίες που βρίσκονται σε τυχαίες θέσεις στο ταμπλό και μπορούν να φωτογραφηθούν από τον παίχτη.

Τα πεδία της:

- Points: οι πόντοι που δίνει η φωτογράφησή της
- Type: ο τύπος της τοιχογραφίας που είναι enum (βλ. σελ 9)
- Constructor που της αρχικοποιεί με τον τύπο της
- Getters για τους πόντους και τον τύπο
- Μια toString για πληροφορίες

2.2.2 Η κλάση SpecialFinding



Η `SpecialFinding` κάνει `extend` την `Finding` και αντιπροσωπεύει τα 'special' ευρήματα τα οποία αφορούν ένα παλάτι το καθένα. Τα πεδία:

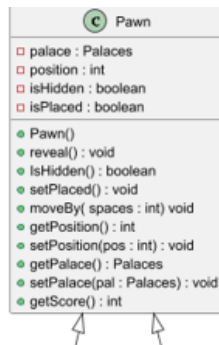
- Οι πόντοι που δίνουν
- Ο τύπος (enum)
- Constructor με τον ειδικό τύπο
- Getters για πόντους και τύπο
- Μια `toString` για πληροφορίες

2.2.3 Η κλάση SnakeGoddess



Η `SnakeGoddess` κάνει `extend` την `Finding` και δεν προσθέτει κάτι πέρα από μια προαιρετική `toString`. Αφορά τα αγαλματάκια που βρίσκονται σε τυχαίες θέσεις.

b. Η κλάση Pawn



Η Pawn αφορά ένα πιόνι στο ταμπλό του παιχνιδιού. Ένα πιόνι μπορεί να βρίσκεται πάνω σε ένα μονοπάτι ή να μην έχει ξεκινήσει καθόλου. Ο κάθε παίχτης έχει ένα πιόνι για κάθε μονοπάτι.

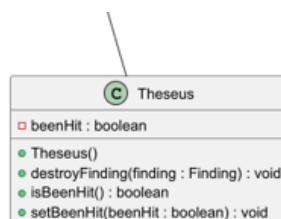
- Palace, το μονοπάτι στο οποίο ανήκει το πιόνι
- Η θέση του στο μονοπάτι
- isHidden: αν είναι κρυμμένο από τον αντίπαλο
- Constructor που αρχικοποιεί ένα πιόνι
- Η reveal () το αποκαλύπτει στον αντίπαλο
- isHidden () getter που επιστρέφει true ή false
- setters και getters για position και palace
- moveBy(spaces) που κινεί το πιόνι

Σημείωση: Θα μπορούσα κάλλιστα να αρχικοποιώ το πιόνι με το παλάτι όπου ανήκει και να μην χρησιμοποιώ setter αφού είναι λίγο αχρείαστο.

i. Η κλάση Archeologist:

Η κλάση αυτή αφορά $\frac{3}{4}$ πιόνια και είναι ένας αρχαιολόγος. Δεν προσθέτει κάτι στην Pawn την οποία κληρονομεί.

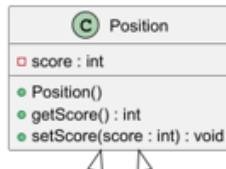
ii. Η κλάση Theseus:



Κάθε παίχτης έχει από έναν Θησέα ο οποίος μπορεί να καταστρέψει ένα εύρημα. Επιπλέον όταν ο αντίπαλος επιτίθεται με κάρτα Μινώταυρου σε

έναν Θησέα, δεν αλλάζει θέση, αλλά δεν μπορεί να χρησιμοποιηθεί στο επόμενο γύρο. Γι' αυτό υπάρχουν και οι κατάλληλες μέθοδοι.

c. Η abstract κλάση Position:

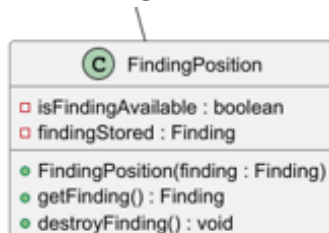


Η Position αφορά μια θέση σε ένα μονοπάτι. Η θέση έχει ένα σκορ καθώς και τους κατάλληλους setters/getter. Επίσης έχει έναν constructor, ο οποίος τώρα που το σκέφτομαι ίσως είναι λίγο περιττός. Παρακάτω θα δούμε δυο κλάσεις που την κληρονομούνε.

i. Η κλάση SimplePosition:

Η SimplePosition είναι μια απλή θέση χωρίς εύρημα και κληρονομεί την Position χωρίς να προσθέτει κάτι.

ii. Η κλάση FindingPosition:



Η θέση αυτή περιέχει ένα εύρημα, κληρονομεί την Position και γι' αυτό έχει:

- Boolean που δείχνει ότι το εύρημα είναι εκεί και δεν το έχει πάρει κάποιος, ή δεν έχει καταστραφεί
- Μεταβλητή τύπου Finding με το εύρημα
- Constructor για τη δημιουργία θέσης με εύρημα
- Getter για το εύρημα
- Συνάρτηση για την καταστροφή του ευρήματος από πιόνι Θησέα

d. Η κλάση Player:



Η κλάση αυτή αντιπροσωπεύει έναν παίχτη στο παιχνίδι. Οι μεταβλητές:

- Το όνομα του παίχτη
- Το σκορ του παίχτη
- Έναν πίνακα με τα ευρήματα που έχει συλλέξει
- Το 'χέρι' του παίχτη
-

Οι μέθοδοι:

- Απλός constructor
- Getters για name, score και lastCardPlayed
- chooseCard: getter για την κάρτα στη θέση index
- setters, getters για ευρήματα
- υπολογισμός σκορ - calculateScore
- κίνηση πιονιού – movePawn
- getPawnForPath – επιστρέφει το πιόνι για ένα μονοπάτι
- addScore για να προσθέσεις score στον παίχτη

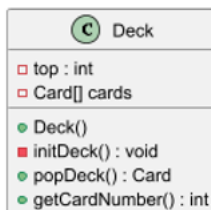
ε. Η κλάση Board:



Η κλάση αφορά το ταμπλό του παιχνιδιού. Συγκεκριμένα έχει τις εξής μεταβλητές:

- Έναν πίνακα 4X9 με τα μονοπάτια των 9 θέσεων τύπου Position (βλ. σελ. 8)
- Ένα αντικείμενο τύπου Deck (βλ. σελ. 10) που είναι οι 100 κάρτες του παιχνιδιού
- Τον πίνακα 4X1 με τις τελευταίες κάρτες που έπαιξε κάθε παίχτης
- Πίνακα 9X1 με τους πόντους κάθε θέσης σε ένα μονοπάτι
- Απλό constructor
- fillPaths – γεμίζει τα μονοπάτια με ευρήματα
- shuffleFindings – κάνει shuffle τα ευρήματα για να μπουν τυχαία
- Συνάρτηση για γέμισμα των θέσεων με ευρήματα
- Απόρριψη κάρτας, είτε επειδή χρησιμοποιήθηκε, είτε επειδή την απέρριψε παίχτης

f. Η κλάση Deck:

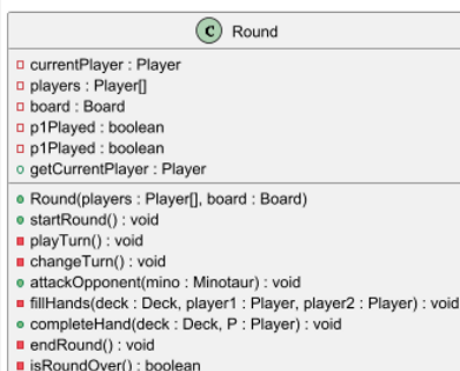


Η Deck αφορά τη στοίβα με τις κάρτες που μοιράζονται στους παίχτες. Όταν η στοίβα αδειάσει, το παιχνίδι τελειώνει. Η στοίβα υλοποιείται με πίνακα.

Τα πεδία της κλάσης είναι μια μεταβλητή για το `top` και ο πίνακας με τις κάρτες, επίσης ένας constructor που όταν καλείται χρησιμοποιεί την `initDeck` για να αρχικοποιήσει τη στοίβα/ πίνακα.

Μια μέθοδος ελέγχει αν η στοίβα είναι άδεια, δηλαδή αν το `top` είναι 0.

g. Η κλάση Round:



Η Round ρυθμίζει έναν γύρο του παιχνιδιού. Ο γύρος περιλαμβάνει: Παίξιμο (ή απόρριψη) μιας κάρτας από κάθε παίχτη και έλεγχος αν το παιχνίδι έχει τελειώσει. Αν όχι προχωράμε σε επόμενο γύρο.

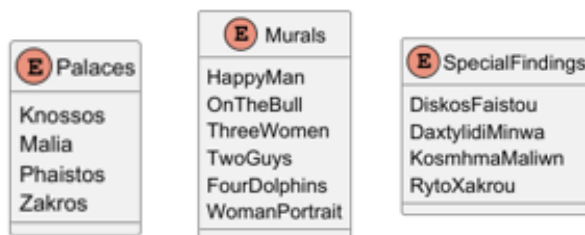
Οι μεταβλητές της κλάσης είναι οι ακόλουθες:

- Μεταβλητή τύπου Player για αποθηκεύει τον παίχτη που παίζει αυτή τη στιγμή
- Πίνακα με τους δύο παίκτες (πιθανόν αχρείαστος)
- Το ταμπλό του παιχνιδιού
- Boolean μεταβλητές για το αν έπαιξαν οι παίκτες

Οι μέθοδοι είναι οι:

- Getter για τον currentPlayer (πρώτη μεταβλητή παραπάνω)
- Constructor που φτιάχνει ένα γύρο με δύο παίκτες και ένα ταμπλό
- Μέθοδοι startRound, changeTurn, endRound που ρυθμίζουν ένα γύρο
- Μέθοδος isRoundOver που επιστρέφει true μόνο αν έπαιξαν και οι δύο παίκτες
- Μέθοδος fillHands που γεμίζει τα χέρια των παιχτών

h. Τα Enums



Τα τρία enum types που χρησιμοποιούνται μέσα στις κλάσεις και αφορούν τα παλάτια, τις τοιχογραφίες και τα ειδικά ευρήματα.

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller



Το controller αποτελείται από μια κλάση που περιέχει μεταβλητές και μεθόδους για τον έλεγχο του παιχνιδιού, όπως μέθοδο για την έναρξη και για την κίνηση των πιονιών. Πιο συγκεκριμένα:

- Πίνακας με τους δύο παίκτες

- Board board, το ταμπλό
- Round – ο γύρος
- Constructor που θα ξεκινάει το παιχνίδι
- createPlayers – αρχικοποίηση παιχτών
- shareCards – μοίρασμα καρτών
- setUpListeners – για να μπουν listeners στις κάρτες και στα υπόλοιπα κουμπιά
- μέθοδοι που λαμβάνουν δράση όταν πατιέται το δεξί ή το αριστερό κλικ
- playMusic – δε λειτουργεί
- determineWinner – για να βγει ο νικητής
- updateUI – ανανέωση όλων των γραφικών

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο view όντας το πιο απλό πακέτο στην υλοποίησή του περιέχει τις εξής τρεις κλάσεις:

1. GraphicUI, η κύρια κλάση που χειρίζεται το παράθυρο του παιχνιδιού και όσα συμβαίνουν μέσα σε αυτό.
2. FindingWindow, ένα παράθυρο τύπου JFrame που θα εμφανίζεται όταν ο χρήστης πατήσει το κουμπί για να δει τα ευρήματά του.
3. FindingInfo, ένα παράθυρο τύπου JFrame που θα εμφανίζεται όταν ο χρήστης πατήσει το κουμπί για να δει πληροφορίες για ένα εύρημα τύπου SpecialFinding.

```
private JPanel player1Panel; //top 2
private JPanel player2Panel; //bottom
private JPanel boardPanel; //middle
private JPanel path1Panel; //inside
private JPanel path2Panel; //inside
private JPanel path3Panel; //inside
private JPanel path4Panel; //inside
private JLabel deck; //contains
private JLabel[][] positions; //to place
private JLabel points; //will be
private JButton discard; //to discard
private JLabel[][] pawns; //4 pawns

private JLabel[] player1Hand; //8 field
private JLabel[] player2Hand; no usages
private JButton[] player1CardSelect;
private JButton[] player2CardSelect; no usages
private JLabel player1Score; // the score
private JLabel player2Score; no usages
private JButton player1Finding; no usages
private JButton player2Finding; no usages
private JCheckBox player1Discard; // check
private JCheckBox player2Discard; no usages
private JLabel[] player1LastCardPlayed;
private JLabel[] player2LastCardPlayed;

private JButton finding1Info; no usages
private JButton finding2Info; no usages
private JButton finding3Info; no usages
private JButton finding4Info; no usages
```

Αριστερά φαίνονται οι μεταβλητές που δηλώθηκαν και θα αποτελέσουν το κύριο κομμάτι του παιχνιδιού.

Ανάμεσά σε αυτές έχουμε τρία Panels, 2 για τους παίχτες και ένα για το ταμπλό, έναν πίνακα από JLabels για τις θέσεις των 4 μονοπατιών, κουμπιά για επιλογή καρτών κ.ά..

Όταν ένας παίχτης στη σειρά του θέλει να απορρίψει μια κάρτα, θα κάνει check στο κατάλληλο checkbox και στη συνέχεια θα επιλέγει την κάρτα.

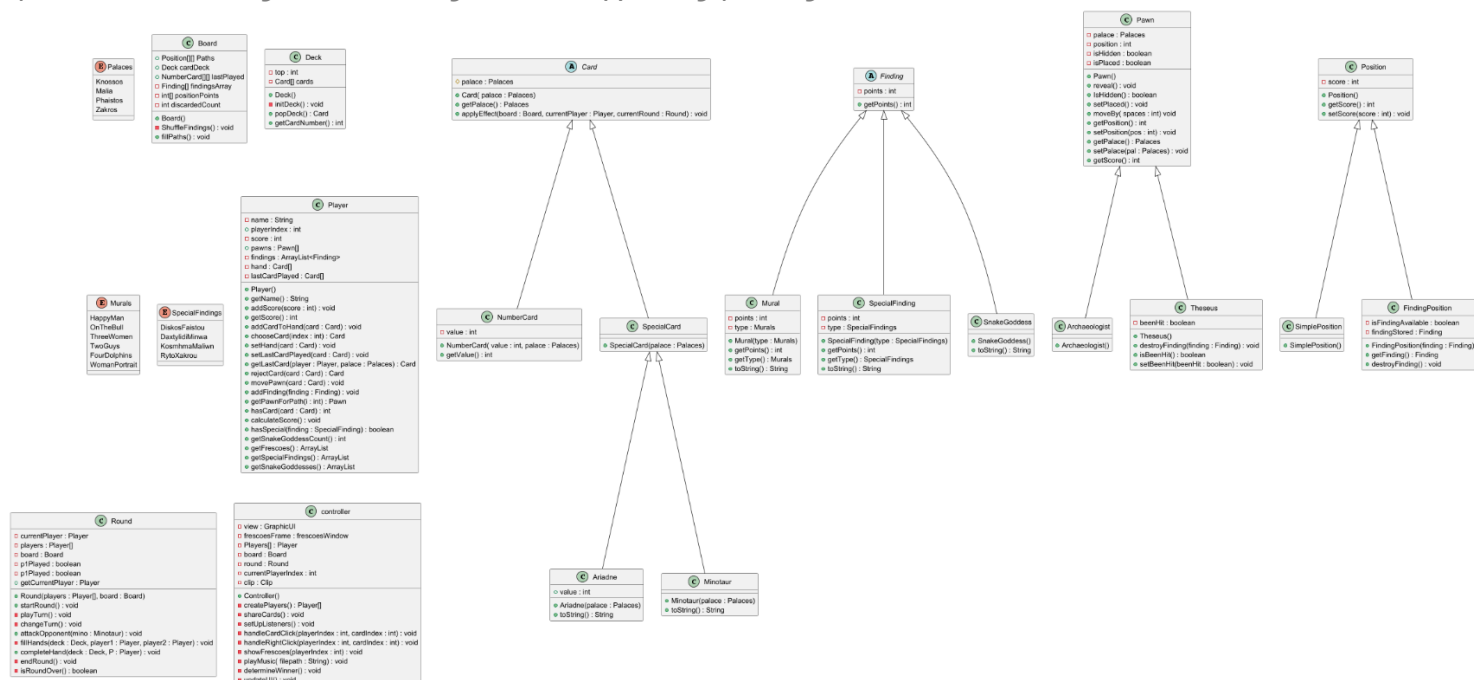
Επιπλέον, μια JLabel θα μπορούσε να αναφέρει ποιανού παίχτη είναι η σειρά.

Όσων αφορά τις μεθόδους της κλάσης GraphicUI:

- constructor που αρχικοποιεί το παράθυρο
- initMenuBar που φτιάχνει την πάνω μπάρα
- initPanels που φτιάχνει τα κύρια Panels του παιχνιδιού
- populateBoard που τοποθετεί τα εικονίδια των μονοπατιών
- setPawn που τοποθετεί την εικόνα ενός πιονιού σε μια θέση
- μέθοδοι που κάνουν update διάφορα γραφικά
- enable-disable buttons για τις κάρτες των παιχτών
- και η selectNames που εμφανίζει ένα JOptionPane για να πάρει τα ονόματα των παιχτών

5. Η αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Παρακάτω υπάρχει μια πλήρης εικόνα του UML διαγράμματος όπου φαίνονται όλες οι κλάσεις και οι σχέσεις μεταξύ αυτών.



6. Junit tests:

Για τις κλάσεις Board, Card, Finding, Pawn, Player φτιάχτηκαν Junit tests που ελέγχουν με assert τις βασικές λειτουργίες τους. Για παράδειγμα στην κλάση Board ελέγχεται ότι τα μονοπάτια αποτελούνται από το σωστό τύπο θέσης (FindingPosition ή SimplePosition). Στην κλάση Pawn ελέγχεται αν ο constructor κάνει initialize ένα κρυμμένο πιόνι, αν η reveal() το εμφανίζει και αν μια κίνηση που το βγάζει εκτός πίνακα κάνει τη θέση το ίση με 10.

7. Συμπεράσματα

A. Αλλαγές:

Το τελικό πρότζεκτ ήταν κάτι αρκετά διαφορετικό και αυτό οφείλεται στο ότι πολλές λειτουργίες του παιχνιδιού μεταφέρθηκαν από την κλάση Round στο Controller.

Επίσης προστέθηκαν αρκετές μέθοδοι όπως η applyEffect στην κλάση Card και η getPawnForPath στην Player.

B. Τι λείπει:

Η μουσική δεν παίζει, τα ευρήματα δεν λειτουργούν σωστά, και οι πόντοι μετρούνται λάθος.

Γ. Προβλήματα κατά την εργασία:

Το view και κυρίως η τοποθέτηση των πλακιδίων στα μονοπάτια ήθελε δουλειά. Επίσης επειδή ήταν η πρώτη φορά που έφτιαξα κάτι με MVC αρχιτεκτονική, ήταν αρχικά λίγο μπερδεμένο το σχέδιο στη φάση Α σχετικά με το τί δουλειές αναλαμβάνει το Controller.