

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2024-2025**

*Εισαγωγή*

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

**Περιεχόμενα**

**1.** **Εισαγωγή 1**

**2.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 1**

**3.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller 1**

**4.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου View 2**

**5.** **Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 2**

**6.** **Λειτουργικότητα (Β Φάση) 2**

**7.** **Συμπεράσματα 2**

* Εισαγωγή

Γραπτή Αναφορά: Πακέτο model

Το πακέτο model αποτελεί το θεμέλιο του project και περιέχει τις βασικές κλάσεις που αναπαριστούν τα δεδομένα και τις δομές του παιχνιδιού. Ακολουθεί μια σύντομη περιγραφή των κλάσεων του πακέτου:

1. Archaeologist

Επεκτείνει την κλάση Pawn.

Περιέχει τη μέθοδο openBox για την επεξεργασία κουτιών ευρημάτων.

2. AriadneCard

Χρησιμοποιείται για την καθοδήγηση πιόνιων.

Περιλαμβάνει μέθοδο επαλήθευσης δυνατότητας παιξίματος.

3. Board

Αναπαριστά το ταμπλό του παιχνιδιού.

.

4. Card

Βασική κλάση για όλες τις κάρτες.

Παρέχει κοινές λειτουργίες για εξειδικευμένες κάρτες.

5. Deck

Διαχειρίζεται την τράπουλα.

Περιλαμβάνει λειτουργίες ανακατέματος, διανομής, και ανανέωσης καρτών.

6. Finding

Αναπαριστά ευρήματα στο παιχνίδι.

Περιλαμβάνει ιδιότητες όπως όνομα, τύπος και σπανιότητα.

7. FrescoFinding

Επεκτείνει την Finding.

Ειδικεύεται σε τοιχογραφίες με πρόσθετους πόντους.

8. MinotaurCard

Αντιπροσωπεύει κάρτες του Μινώταυρου με λειτουργίες επίθεσης.

9. NumberCard

Αναπαριστά αριθμητικές κάρτες για κινήσεις πιόνιων.

10. Path

Αναπαριστά μονοπάτια του ταμπλό.

Παρέχει διαχείριση θέσεων, πιόνιων, και ευρημάτων.

11. Pawn

Αναπαριστά τα πιόνια.

Περιλαμβάνει ιδιότητες για θέση και παίκτη.

12. Player

Αναπαριστά έναν παίκτη.

Παρέχει διαχείριση καρτών και πόντων.

13. Position

Αναπαριστά θέση στο ταμπλό.

Διαχειρίζεται πιόνια και ευρήματα.

14. RareFinding

Επεκτείνει την Finding.

Αναπαριστά σπάνια ευρήματα με μεγαλύτερη αξία.

15. StatueFinding

Επεκτείνει την Finding.

Αναπαριστά ευρήματα αγαλμάτων.

16. Theseus

Επεκτείνει την Pawn.

Περιλαμβάνει τη μέθοδο destroyBox για καταστροφή κουτιών ευρημάτων.

##### 17. playMusic

* Παίζει το αρχείο μουσικής που αντιστοιχεί στον τρέχοντα παίκτη.
* Σημαντικά Στοιχεία:
  + Διακόπτει τυχόν υπάρχουσα αναπαραγωγή.
  + Φορτώνει και αναπαράγει το νέο αρχείο μουσικής.

Σχεδίαση και Κλάσεις του Πακέτου View

Το πακέτο view είναι υπεύθυνο για τη διαχείριση της γραφικής διεπαφής χρήστη (GUI) και περιλαμβάνει την κλάση GameView. Ακολουθεί μια περιγραφή των βασικών συναρτήσεων της κλάσης:

1. GameView

Αναπαριστά το κύριο παράθυρο του παιχνιδιού.

Διαχειρίζεται μονοπάτια, κάρτες, και την ετικέτα τρέχοντος παίκτη.

Βασικές Συναρτήσεις

1. Constructor GameView()

Αρχικοποιεί το κύριο παράθυρο.

Δημιουργεί χάρτη pathLabelMap για μονοπάτια.

Ρυθμίζει εικόνες φόντου και panel παικτών.

2. setController(GameController controller, List<Player> players)

Συνδέει τον controller με το view.

Ρυθμίζει listeners στα κουμπιά καρτών.

3. updateBoard(Board board)

Ενημερώνει το ταμπλό με βάση το παιχνίδι.

Φορτώνει εικόνες για μονοπάτια, θέσεις, και ευρήματα.

4. updateCurrentPlayer(String playerName)

Ενημερώνει την ετικέτα του τρέχοντος παίκτη.

5. displayGameOverMessage()

Εμφανίζει μήνυμα "Game Over".

6. updatePlayerHandPanel(Player player, List<Card> updatedCards)

Ενημερώνει τις κάρτες του παίκτη στο GUI.

7. enablePlayerInteraction(List<Player> players, int playerId, GameController controller)

Ενεργοποιεί την αλληλεπίδραση για τον ενεργό παίκτη.

8. disablePlayerInteraction()

Απενεργοποιεί την αλληλεπίδραση για όλους τους παίκτες.

9. addPath(String pathName, int index)

Δημιουργεί γραφικές αναπαραστάσεις μονοπατιών.

10. loadAndResizeImage(String filename, int width, int height)

Φορτώνει και αλλάζει μέγεθος εικόνων.

11. createPlaceholderImage(int width, int height)

Δημιουργεί placeholder εικόνες.

Πακέτο Controller

Κλάση GameController

Βασικές Μέθοδοι

1. Constructor GameController(...)

Αρχικοποιεί βασικά στοιχεία του παιχνιδιού.

Συνδέει τον controller με το view.

2. startGame()

Ξεκινά το παιχνίδι.

Αρχικοποιεί τα στοιχεία του παιχνιδιού.

3. playGameLoop()

Τρέχει τη βασική ροή του παιχνιδιού.

4. handleCardClick(...)

Διαχειρίζεται την επιλογή κάρτας από τον παίκτη.

5. initializePawns(Player player)

Δημιουργεί πιόνια για κάθε παίκτη.

6. placePawnsOnPaths(...)

Τοποθετεί πιόνια στα μονοπάτια.

7. initializeDeckData(Deck deck)

Αρχικοποιεί την τράπουλα.

8. switchPlayerTurn()

Εναλλάσσει τη σειρά παικτών.

9. isGameOver()

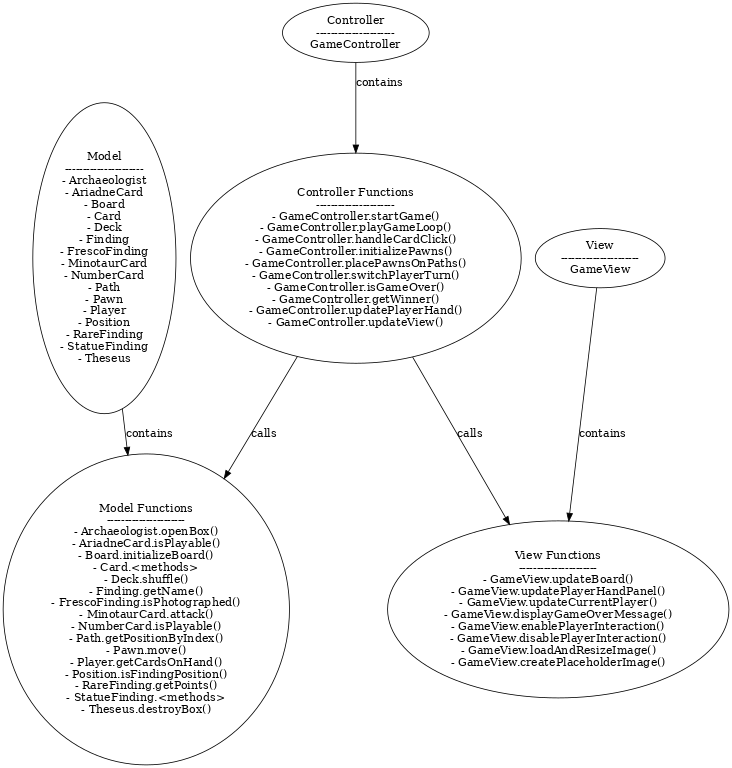
Ελέγχει αν το παιχνίδι τελείωσε.

10. getWinner()

Επιστρέφει τον νικητή του παιχνιδιού.

Αυτή η δομή παρέχει μια σαφή εικόνα για τα πακέτα model, view, και controller, περιγράφοντας τις κύριες λειτουργίες τους και την αλληλεπίδραση μεταξύ τους.

* Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



* Λειτουργικότητα (Β Φάση)

Σε αυτή την φάση κατάφερα να υλοποιήσω τα περισσότερα ερωτήματα ,πλην των ερωτημάτων που αφορούν την συλλογή των ευρημάτων.

* Συμπεράσματα

Γενικά το πρότζεκτ ήταν αρκετά απαιτητικό ,δυσκολεύτηκα σε ορισμένα σημεία καθώς δεν είχα παρόμοιες εμπειρίες σε τέτοια πρότζεκτ ,ωστόσο κατάφερα να το υλοποιήσω σε ένα ικανοποιητικό επίπεδο.

Σε σχέση με τη Α φάση υπήρξαν αρκετές αλλαγές και προσθήκες ωστόσο οι περισσότερες και πιο βασικές συναρτήσεις από την πρώτη φάση δεν υπέστησαν ιδιαίτερες αλλαγές. Το βασικό πρόβλημα που αντιμετώπισα ήταν η λειτουργικότητα της συλλογής των ευρημάτων και δεν κατάφερα να το διορθώσω. Κατ’ επέκταση οι συναρτήσεις που αφορούν το σκορ των παικτών και τον νικητή ,αν και υλοποιήθηκαν (και σε στα JuniTests) δεν γινόταν να αξιοποιηθούν στο πρόγραμμα.