

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2024-2025**

HY-252 Project

Phase\_A

Εισαγωγή

ΝΙΚΟΛΑΟΣ-ΝΕΚΤΑΡΙΟΣ ΧΡΟΝΗΣ

ΑΜ5174

3/12/2024

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

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

Αυτό το project υλοποιεί μια παραλλαγή του επιτραπέζιου παιχνιδιού ‘Χαμένες Πόλεις’. Αυτή η φάση επικεντρώνεται στην σχεδίαση του παιχνιδιού και όχι ιδιαίτερα στην υλοποίηση. Ως αρχιτεκτονική θα χρησιμοποιηθεί το μοντέλο MVC και παρακάτω θα ακολουθήσουν λεπτομέρειες του σχεδιασμού όπως οι κλάσεις που θα χρησιμοποιηθούν σε κάθε στάδιο ,τι θα περιέχουν αυτές (πχ χαρακτηριστικά και μεθόδους) ,σε τι χρησιμεύουν ,καθώς και λεπτομερή διαγράμματα UML για καλύτερη κατανόηση.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

1. **Εισαγωγή**

* **Περιγραφή του Model:**
  + Το Model είναι υπεύθυνο για τη διαχείριση όλων των δεδομένων και της λογικής του παιχνιδιού.
  + Περιλαμβάνει τις κλάσεις για τις κάρτες, τους παίκτες, τα πιόνια, τα μονοπάτια, τις θέσεις, τα αρχαιολογικά ευρήματα, και το ταμπλό.
  + Ακολουθεί το πρότυπο **MVC**, οπότε το Model είναι πλήρως ανεξάρτητο από την παρουσίαση (View) και τη ροή του παιχνιδιού (Controller).

**2. Ανάλυση Κλάσεων και Υποκλάσεων**

Διαχωρίζεις την περιγραφή σε ενότητες, μία για κάθε βασική κλάση.

**2.1 Κλάση Card**

* **Περιγραφή:**
  + Η κλάση Card είναι η βασική κλάση για όλες τις κάρτες του παιχνιδιού.
  + Διαχωρίζεται σε υποκλάσεις (NumberCard, AriadneCard, MinotaurCard) για να διαχειριστούμε διαφορετικούς τύπους καρτών με μοναδικές λειτουργίες.
* **Λόγος διαχωρισμού σε υποκλάσεις:**
  + Οι αριθμητικές κάρτες (NumberCard) χρησιμοποιούνται για την κίνηση.
  + Οι κάρτες Μίτου Αριάδνης (AriadneCard) προσφέρουν ειδική κίνηση δύο θέσεων.
  + Οι κάρτες Μινώταυρου (MinotaurCard) επιτρέπουν επίθεση στον αντίπαλο.
* **Μέθοδοι :**
  + isPlayable(Card previousCard):
    - Ελέγχει αν η κάρτα μπορεί να παιχτεί με βάση τους κανόνες του παιχνιδιού.
  + attack(Player opponent, Path path):
    - Χρησιμοποιείται από τις κάρτες Μινώταυρου για να εκτελέσουν επίθεση (στο Controller).

**2.2 Κλάση Player**

* **Περιγραφή:**
  + Αναπαριστά έναν παίκτη του παιχνιδιού. Κάθε παίκτης έχει όνομα, κάρτες, πιόνια, σκορ, και αρχαιολογικά ευρήματα.
* **Μέθοδοι :**
  + addScore(int points):
    - Προσθέτει πόντους στο σκορ του παίκτη.
  + playCard(Card card, Path path):
    - Ο παίκτης παίζει μια κάρτα και κινεί ένα πιόνι (στο Controller).
  + drawCard(Deck deck):
    - Τραβάει μια κάρτα από την τράπουλα.
  + hasAvailablePawn(Path path):
    - Ελέγχει αν ο παίκτης έχει διαθέσιμο πιόνι για ένα μονοπάτι.

**2.3 Κλάση Pawn και Υποκλάσεις**

* **Περιγραφή:**
  + Η κλάση Pawn αναπαριστά ένα πιόνι. Έχει δύο υποκλάσεις:
    - Archaeologist: Μπορεί να "ανοίξει" ένα κουτί για να πάρει ένα εύρημα.
    - Theseus: Μπορεί να "καταστρέψει" ένα κουτί.
* **Μέθοδοι :**
  + move(int steps):
    - Μετακινεί το πιόνι κατά συγκεκριμένα βήματα.
  + reveal():
    - Αποκαλύπτει το πιόνι όταν απαιτείται.
  + openBox(FindingPosition position) (Archaeologist):
    - Ανοίγει ένα κουτί σε συγκεκριμένη θέση και συλλέγει εύρημα.
  + destroyBox(FindingPosition position) (Theseus):
    - Καταστρέφει ένα κουτί.

**2.4 Κλάση Path**

* **Περιγραφή:**
  + Αναπαριστά ένα μονοπάτι που συνδέει ένα ανάκτορο.
  + Αποτελείται από 9 θέσεις (Position), κάποιες από τις οποίες περιέχουν ευρήματα.
* **Μέθοδοι :**
  + getNextPosition(Pawn pawn):
    - Επιστρέφει την επόμενη διαθέσιμη θέση για το πιόνι.
  + isCompleted():
    - Επιστρέφει αν το μονοπάτι έχει ολοκληρωθεί.

**2.5 Κλάση Position και Υποκλάσεις**

* **Περιγραφή:**
  + Αναπαριστά μια θέση σε ένα μονοπάτι. Υπάρχουν δύο τύποι:
    - SimplePosition: Δεν περιέχει εύρημα.
    - FindingPosition: Περιέχει ένα εύρημα.
* **Μέθοδοι :**
  + getScore():
    - Επιστρέφει τη βαθμολογία της θέσης.
  + revealFinding() (FindingPosition):
    - Αποκαλύπτει το εύρημα της θέσης.

**2.6 Κλάση Board**

* **Περιγραφή:**
  + Αναπαριστά το ταμπλό του παιχνιδιού. Περιέχει τα μονοπάτια, την τράπουλα και τα σπάνια ευρήματα.
* **Μέθοδοι :**
  + initializeBoard():
    - Αρχικοποιεί τα μονοπάτια, τις θέσεις και την τράπουλα.
  + getPath(String palace):
    - Επιστρέφει το μονοπάτι για ένα συγκεκριμένο ανάκτορο.

**2.7 Κλάση Deck**

* **Περιγραφή:**
  + Διαχειρίζεται την τράπουλα και τη στοίβα απόρριψης.
* **Μέθοδοι :**
  + shuffle():
    - Ανακατεύει τις κάρτες.
  + draw():
    - Τραβάει μια κάρτα από την τράπουλα.
  + discard(Card card):
    - Προσθέτει μια κάρτα στη στοίβα απόρριψης.

A diagram of a network

Description automatically generated

\*Στο διάγραμμα UML απεικονίζονται οι σχέσεις μεταξύ των κλάσεων και των υποκλάσεων στο package του Model. Σε κάθε κλάση και υποκλάση αναφέρονται και τα αντίστοιχα χαρακτηριστικά και μέθοδοι που είναι απαραίτητα ,καθώς και ο τύπος δεδομένων τους.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

\* **setupBoard()**

* **Χρησιμότητα:** Αρχικοποιεί το παιχνίδι, ορίζοντας το board με paths, findings, και την τράπουλα (deck).
* **Λειτουργικότητα:** Καλεί μεθόδους από το Model (π.χ., initializeBoard()) για να θέσει τα δεδομένα.

\* **randomStart()**

* **Χρησιμότητα:** Επιλέγει τυχαία τον παίκτη που θα ξεκινήσει πρώτος.
* **Λειτουργικότητα:** Χρησιμοποιεί την κλάση Random για να αποφασίσει ποιος παίκτης θα παίξει πρώτος.

\* **getCurrentPlayer()**

* **Χρησιμότητα:** Επιστρέφει τον παίκτη που παίζει αυτή τη στιγμή.
* **Λειτουργικότητα:** Χρησιμοποιείται από άλλες μεθόδους για να γνωρίζουν τον ενεργό παίκτη.

\* **turn()**

* **Χρησιμότητα:** Υλοποιεί τη λογική του γύρου για τον τρέχοντα παίκτη.
* **Λειτουργικότητα:** Περιλαμβάνει την αλληλουχία ενεργειών που εκτελεί ο παίκτης κατά τη διάρκεια του γύρου του.

\* **validMove(Pawn pawn, Card card)**

* **Χρησιμότητα:** Ελέγχει αν η κίνηση ενός pawn είναι έγκυρη, βάσει των κανόνων και της κάρτας που χρησιμοποιείται.
* **Λειτουργικότητα:** Επικοινωνεί με το Model για να ελέγξει τη θέση στο board και τη συμβατότητα της κίνησης.

\* **nextTurn()**

* **Χρησιμότητα:** Εναλλάσσει τον ενεργό παίκτη στον επόμενο.
* **Λειτουργικότητα:** Διαχειρίζεται τη μετάβαση από τον έναν παίκτη στον άλλον.

\* **Winner()**

* **Χρησιμότητα:** Ελέγχει αν κάποιος παίκτης έχει κερδίσει.
* **Λειτουργικότητα:** Θα καλεί μεθόδους του Model για να αξιολογήσει τις συνθήκες νίκης.
* **isGameOver()**
* **Χρησιμότητα:** Επιστρέφει true αν το παιχνίδι έχει ολοκληρωθεί.
* **Λειτουργικότητα:** Βασίζεται σε ελέγχους του Model.

\* **movePawn(Pawn pawn, int steps)**

* **Χρησιμότητα:** Μετακινεί ένα pawn κατά τον αριθμό βημάτων που καθορίζεται.
* **Λειτουργικότητα:** Χρησιμοποιεί δεδομένα από το Model για να ενημερώσει τη θέση του pawn στο board.

\* **playCard(Pawn pawn, Card card)**

* **Χρησιμότητα:** Εφαρμόζει την επίδραση μιας κάρτας σε ένα pawn.
* **Λειτουργικότητα:** Αλληλεπιδρά με το Model για να επιβεβαιώσει την ισχύ της κάρτας και να ενημερώσει την κατάσταση του παιχνιδιού.

\* **countScore(int points)**

* **Χρησιμότητα:** Υπολογίζει το σκορ των παικτών και καθορίζει τον νικητή.
* **Λειτουργικότητα:** Καλεί τις μεθόδους getScore() του κάθε παίκτη για τη σύγκριση των πόντων.

A screenshot of a computer game

Description automatically generated

\*Στη Β’ φάση θα υλοποιηθούν οι μέθοδοι των κλάσεων όσων αφορά την λειτουργικότητα του παιχνιδιού. Ενδέχεται να χρειαστούν αλλαγές και νέες προσθήκες μεθόδων ,χαρακτηριστικών και κλάσεων-υποκλάσεων.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο **View** είναι υπεύθυνο για την εμφάνιση του γραφικού περιβάλλοντος του παιχνιδιού και την αλληλεπίδραση με τον χρήστη. Ο στόχος του είναι να αναπαραστήσει οπτικά την κατάσταση του παιχνιδιού και να επιτρέψει στους παίκτες να δώσουν τις εντολές τους με έναν φιλικό και ευχάριστο τρόπο.

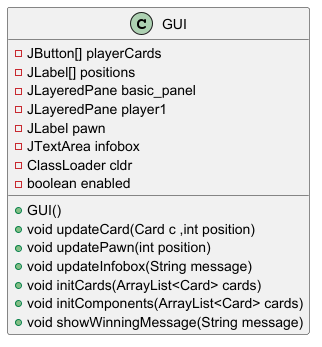
**Κλάσεις και Χαρακτηριστικά**

1. **Κύρια Κλάση: GUI**
   * Η κλάση GUI θα επεκτείνει τη JFrame για να παρέχει το κύριο γραφικό παράθυρο του παιχνιδιού.
   * Θα περιλαμβάνει τα βασικά στοιχεία του γραφικού περιβάλλοντος:
     + **Κουμπιά (JButton):**
       - Αναπαριστούν τις κάρτες του παίκτη.
     + **Ετικέτες (JLabel):**
       - Αναπαριστούν τις θέσεις στο board και το pawn του παίκτη.
     + **Πεδίο Πληροφοριών (JTextArea):**
       - Εμφανίζει μηνύματα και πληροφορίες για το παιχνίδι.
     + **Πάνελ (JLayeredPane):**
       - Χρησιμοποιούνται για τη διάταξη των γραφικών στοιχείων.

**Μέθοδοι**

Οι βασικές μέθοδοι που θα υλοποιηθούν περιλαμβάνουν:

1. **updateCard(Card c, int position)**
   * Ενημερώνει την εικόνα μιας κάρτας στον πίνακα καρτών του παίκτη.
2. **updatePawn(int position)**
   * Μετακινεί το pawn του παίκτη στη νέα του θέση στο board.
3. **updateInfobox(String message)**
   * Ενημερώνει το πεδίο πληροφοριών με το μήνυμα που του δίνεται.
4. **initCards(ArrayList<Card> cards)**
   * Δημιουργεί και εμφανίζει τα κουμπιά για τις κάρτες του παίκτη.
5. **initComponents(ArrayList<Card> cards)**
   * Αρχικοποιεί όλα τα γραφικά στοιχεία της διεπαφής.
6. **WinningMessage(String message)**
   * Εμφανίζει ένα μήνυμα νίκης στον παίκτη μέσω διαλόγου.



## Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Εδώ παρέχεται ένα UML διάγραμμα όπου παρουσιάζει την αλληλεπίδραση μεταξύ των κλάσεων διαφορετικών πακέτων. Πρακτικά είναι πολύ δυσκολο να συμπεριληφθούν όλες οι κλάσεις-υποκλάσεις και μέθοδοι ,συνεπώς υπάρχουν οι μόνο οι βασικότερες και πιο καθοριστικές. Εδώ λοιπόν παρατηρούμε ότι :

-Ο Controller ελέγχει πρακτικά όλο το παιχνίδι ,αρχικοποιεί και διαχειρίζεται την κλάση Board του πακέτου Model.

-Επίσης διαχειρίζεται την κλάση Pawn του πακέτου Model ,διασφαλίζοντας ότι τηρούνται οι κανόνες του παιχνιδιού (πχ έγκυρες κινήσεις των pawns).

-Ο Controller αλληλεπιδρά και με τον παίκτη ,πχ ελέγχοντας την σειρά με την οποία παίζουν οι παίκτες (εναλλάξ) ,αλλά και το ποιος θα ξεκινήσει το παιχνίδι (τυχαία).

-Ο Controller επίσης διαχειρίζεται τις κάρτες του παίκτη ,όπου η κάθε μια έχει διαφορετική συμπεριφορά (πχ οι κάρτες μινώταυρου διαφέρουν σε ιδιότητες από τις απλές αριθμητικές) ,και διασφαλίζει αν ο παίκτης μπορεί να παίξει μία συγκεκριμένη κάρτα με βάση τους κανόνες.

-Τέλος ,το πακέτο View βρίσκεται υπό τις οδηγίες του Controller και είναι υπεύθυνο να εμφανίζει στην οθόνη μηνύματα (πχ μήνυμα για τον νικητή και το τέλος παιχνιδιού) ,καθώς και να εναλλάσσει το backround ,την μουσική ,το board και τις κάρτες του παίκτη.

A diagram of a diagram

Description automatically generated

## Λειτουργικότητα (Β Φάση)

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

## Συμπεράσματα

Σε αυτήν την ενότητα θα γράψετε τα συμπεράσματα σας για την εργασία, τυχόν προβλήματα που συναντήσατε και γενικά ότι άλλο κρίνετε απαραίτητο να αναφερθεί.