

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2023-2024**

Project SORRY

Εισαγωγή

Νικος Λαζαριδης

4922

4 – 12 - 2023

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

Η υλοποίηση του επιτραπέζιου παιχνιδιού "Sorry" σε Java ακολουθεί τον σχεδιασμό πρότυπου Model-View-Controller (MVC).

Το μέρος του Model του προτύπου MVC εκπροσωπείται από κλάσεις όπως η Board, η Deck και η Card. Η κλάση Board μοντελοποιεί το παιχνιδικό ταμπλό και περιλαμβάνει μεθόδους για την αρχικοποίηση του ταμπλό, τον έλεγχο της εγκυρότητας των κινήσεων, τη μετακίνηση των πιόνων και τον καθορισμό του νικητή. Η κλάση Deck αρχικοποιεί και διαχειρίζεται ένα τράπουλα χαρτιών για το παιχνίδι, υποστηρίζοντας μεθόδους για την εξαγωγή χαρτιών, τον έλεγχο αν η τράπουλα είναι άδεια και άλλα. Η διεπαφή Card μοντελοποιεί ένα χαρτί στο παιχνίδι και ορίζει τα χαρακτηριστικά του, καθώς και χρήσιμες μεθόδους.

Το μέρος του Controller του προτύπου MVC εκπροσωπείται από την κλάση Controller. Αυτή η κλάση διαχειρίζεται τη λογική του παιχνιδιού, συμπεριλαμβανομένων των ενεργειών των παικτών και των αλλαγών κατάστασης. Αρχικοποιεί τα χαρτιά, τους παίκτες και το ταμπλό, και χειρίζεται διάφορες παιχνιδικές μεθόδους.

Το μέρος της View του προτύπου MVC είναι υπεύθυνο για την εμφάνιση της κατάστασης του παιχνιδιού στον χρήστη και την χειρισμό της εισόδου του χρήστη

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Στην Α φάση του Project, το πακέτο "model" σχεδιάστηκε να περιλαμβάνει τις βασικές κλάσεις που αντιπροσωπεύουν τα στοιχεία του παιχνιδιού "Sorry". Παρακάτω παρουσιάζεται η σχεδίαση κάθε κλάσης στο πακέτο "model".

Παρακατω αναλυω τα πακετα Board , Card , Player :

Το πακέτο "board" υλοποιεί το ταμπλό του παιχνιδιού "Sorry", το οποίο είναι ένα 16x16 πλέγμα. Περιλαμβάνει τη διεπαφή Square και διάφορες υποκλάσεις που μοντελοποιούν διάφορα είδη τετραγώνων στο ταμπλό.

1. **Διεπαφή Square:**
   * Περιγραφή: Αυτή η διεπαφή μοντελοποιεί ένα γενικό τετράγωνο στο ταμπλό. Κάθε τετράγωνο έχει μία συγκεκριμένη θέση και χρώμα.
   * Μέθοδοι:
     + getPosition(): int: Επιστρέφει τη θέση του τετραγώνου.
     + getColor(): Color: Επιστρέφει το χρώμα του τετραγώνου.
     + setColor(Color color): void: Ορίζει το χρώμα του τετραγώνου.
     + isOccupied(): boolean: Επιστρέφει true αν το τετράγωνο είναι κατειλημμένο.
     + setOccupied(boolean occupied): void: Ορίζει την κατάσταση κατάληψης του τετραγώνου.
2. **StartSquare:**
   * Περιγραφή: Αντιπροσωπεύει ένα τετράγωνο εκκίνησης για έναν παίκτη. Έχει ειδικά χαρακτηριστικά που το καθιστούν απρόσβατο από τον αντίπαλο παίκτη.
3. **SimpleSquare:**
   * Περιγραφή: Αντιπροσωπεύει ένα απλό τετράγωνο στο ταμπλό χωρίς ειδικά χαρακτηριστικά.
4. **SafetyZoneSquare:**
   * Περιγραφή: Αντιπροσωπεύει ένα τετράγωνο ασφαλείας που προστατεύει τα πιόνια από επιθέσεις του αντίπαλου.
5. **HomeSquare:**
   * Περιγραφή: Αντιπροσωπεύει το τετράγωνο στόχου, όπου τα πιόνια πρέπει να φτάσουν για να κερδίσουν.
6. **SlideSquare:**
   * Περιγραφή: Αντιπροσωπεύει ένα τετράγωνο τσουλήθρας που επιτρέπει στα πιόνια να ολισθαίνουν σε μεγαλύτερες αποστάσεις.
7. **StartSlideSquare, InternalSlideSquare, EndSlideSquare:**
   * Περιγραφή: Υποκλάσεις της SlideSquare που αντιπροσωπεύουν την αρχή, το εσωτερικό κομμάτι, και το τέλος μιας τσουλήθρας αντίστοιχα.

Το πακέτο "cards" υλοποιεί το σύνολο των καρτών που χρησιμοποιούνται στο παιχνίδι "Sorry". Περιλαμβάνει τη διεπαφή Card και διάφορες κλάσεις καρτών που υλοποιούν αυτήν τη διεπαφή. Αναλυτικά:

**Διεπαφή Card:**

* + Περιγραφή: Αυτή η διεπαφή ορίζει τις βασικές μεθόδους και χαρακτηριστικά που πρέπει να υλοποιούνται από όλες τις κάρτες στο παιχνίδι "Sorry".
  + Μέθοδοι:
    - **getValue():**
      * Τύπος: Accessor (ανάκτηση).
      * Περιγραφή: Επιστρέφει την αξία της κάρτας.
    - **SetValue(int value):**
      * Τύπος: Transformer (τροποποίηση).
      * Περιγραφή: Ορίζει την αξία της κάρτας με βάση την παρεχόμενη τιμή.
    - **SetColor(Color color):**
    - Τύπος: Transformer (τροποποίηση).
    - Περιγραφή: Ορίζει το χρώμα της κάρτας με βάση τον παρεχόμενο χρωματικό κώδικα.
  + **Color getColor():**
    - Τύπος: Accessor (ανάκτηση).
    - Περιγραφή: Επιστρέφει το χρώμα της κάρτας.
  + **toString():**
    - Τύπος: Accessor (ανάκτηση).
    - Περιγραφή: Επιστρέφει μια συμβολοσειρά που αναπαριστά την κάρτα.
  + **movePawn():**
    - Τύπος: Transformer (τροποποίηση).
    - Περιγραφή: Εκτελεί μια κίνηση σχετική με την κάρτα, όπως τη μετακίνηση ενός πιονιού.

1. **Κλάση Deck:**
   * Περιγραφή: Η κλάση αυτή υλοποιεί τη διεπαφή Card και αντιπροσωπεύει ένα deck καρτών στο παιχνίδι. Περιλαμβάνει μεθόδους για την αρχικοποίηση του deck, ανακάτεμα, και την ανάκτηση καρτών.
2. **Κλάση NumberElevenCard:**
   * Περιγραφή: Αυτή η κλάση επεκτείνει την SimpleNumberCard και αντιπροσωπεύει μια κάρτα με αριθμό "11". Είναι μια απλή αριθμητική κάρτα χωρίς επιπλέον ειδικές λειτουργίες.
3. **Κλάση NumberFourCard:**
   * Περιγραφή: Αυτή η κλάση επεκτείνει την SimpleNumberCard και αντιπροσωπεύει μια κάρτα με αριθμό "4". Έχει ειδική λειτουργία που κάνει το πιόνι να κινείται προς τα πίσω κατά τέσσερις θέσεις.
4. **Κλάση NumberOneCard:**
   * Περιγραφή: Αυτή η κλάση επεκτείνει την SimpleNumberCard και αντιπροσωπεύει μια κάρτα με αριθμό "1". Είναι απλή αριθμητική κάρτα χωρίς επιπλέον ειδικές λειτουργίες.
5. **Κλάση NumberSevenCard:**
   * Περιγραφή: Αυτή η κλάση επεκτείνει την SimpleNumberCard και αντιπροσωπεύει μια κάρτα με αριθμό "7". Είναι απλή αριθμητική κάρτα χωρίς επιπλέον ειδικές λειτουργίες.
6. **Κλάση NumberTenCard:**
   * Περιγραφή: Αυτή η κλάση επεκτείνει την SimpleNumberCard και αντιπροσωπεύει μια κάρτα με αριθμό "10". Είναι απλή αριθμητική κάρτα χωρίς επιπλέον ειδικές λειτουργίες.
7. **Κλάση SimpleNumberCard:**
   * Περιγραφή: Αυτή η κλάση υλοποιεί τη διεπαφή Card και αντιπροσωπεύει μια απλή αριθμητική κάρτα χωρίς ειδικές λειτουργίες.
8. **Κλάση SorryCard:**
   * Περιγραφή: Αυτή η κλάση υλοποιεί τη διεπαφή Card και αντιπροσωπεύει την ειδική κάρτα "Sorry". Έχει μοναδικούς κανόνες και λειτουργίες στο παιχνίδι.

Η κλάση Turn στο πακέτο Model.Turn αντιπροσωπεύει τη σειρά του παιχνιδιού. Διαθέτει ιδιωτικά πεδία για την τρέχουσα ταυτότητα του παίκτη (currentID) και τη σειρά (turn). Η κλάση παρέχει μεθόδους για τον ορισμό, την αύξηση ή τη μείωση της σειράς, καθώς και για την ανάκτηση της τρέχουσας ταυτότητας και της σειράς του παιχνιδιού

Η κλάση Pawn στο πακέτο Model.Player αντιπροσωπεύει ένα πιόνι στο παιχνίδι. Κάθε πιόνι έχει ένα χρώμα (color), μια θέση (position) και μια κατάσταση ενεργού (active). Η κλάση παρέχει μεθόδους για τον ορισμό και την ανάκτηση του χρώματος και της θέσης του πιόνι, καθώς και για την κίνηση του πιόνι προς τα εμπρός ή προς τα πίσω

Η κλάση Player αντιπροσωπεύει έναν παίκτη στο παιχνίδι. Κάθε παίκτης έχει ένα όνομα (name), ένα χρώμα (color), δύο πιόνια (pawns) και μια μεταβλητή (hasPlayed) που δηλώνει αν έχει παίξει. Η κλάση παρέχει μεθόδους για τον ορισμό και την ανάκτηση του ονόματος, του χρώματος και της ταυτότητας του παίκτη, καθώς και για την αρχικοποίηση του παίκτη και την επαναφορά των πιονιών του.

## Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Η κλάση Controller στην υλοποίηση του επιτραπέζιου παιχνιδιού "Sorry" σε Java διαχειρίζεται τη λογική του παιχνιδιού, συμπεριλαμβανομένων των ενεργειών των παικτών και των αλλαγών κατάστασης. Αρχικοποιεί τις κάρτες, τους παίκτες και το ταμπλό, και χειρίζεται διάφορες μεθόδους που σχετίζονται με το παιχνίδι. Η κλάση διαθέτει ιδιωτικά χαρακτηριστικά για δύο παίκτες (P1 και P2), ένα ArrayList παικτών, έναν πίνακα πιονιών, ένα τράπουλα χαρτών, ένα μετρητή fold, ένα μετρητή νικητή, ένα boolean endGame, ένα αντικείμενο Turn και ένα αντικείμενο Board.

Η κλάση Controller περιλαμβάνει τις παρακάτω μεθόδους:

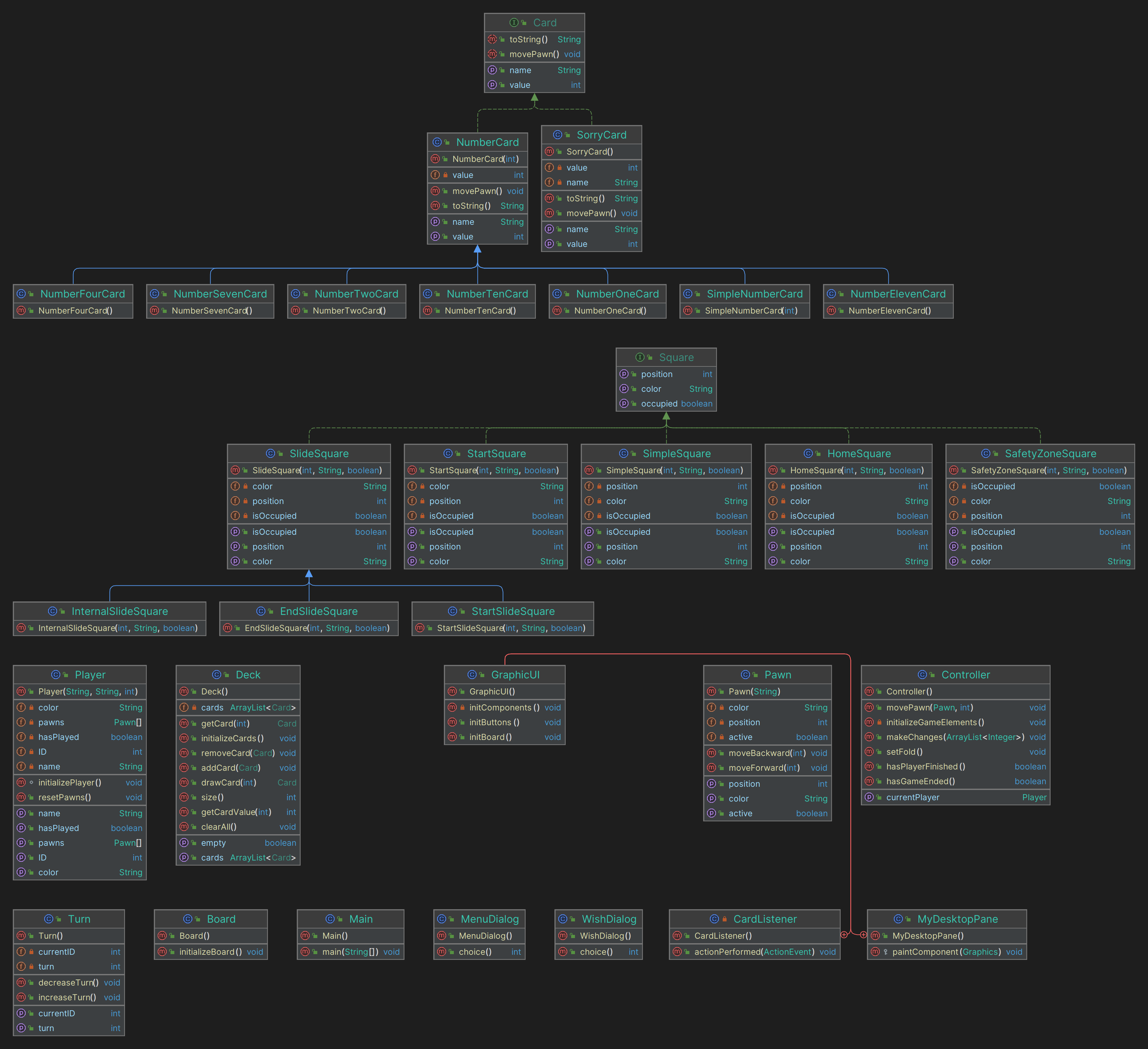
1. **Ο constructor (κατασκευαστής) αρχικοποιεί τα στοιχεία του παιχνιδιού.**
2. **Η μέθοδος initializeGameElements() αρχικοποιεί τις κάρτες, τους παίκτες, τα πιόνια και το ταμπλό.**
3. **Η μέθοδος hasPlayerFinished() ελέγχει εάν και τα δύο πιόνια ενός παίκτη βρίσκονται στο τετράγωνο "home", υποδηλώνοντας ότι το παιχνίδι έχει τελειώσει.**
4. **Η μέθοδος movePawn(Pawn pawn, int newPosition) μετακινεί ένα πιόνι από μια θέση σε μια άλλη στο ταμπλό.**
5. **Η μέθοδος hasGameEnded() καθορίζει εάν το παιχνίδι έχει τελειώσει, είτε λόγω νικητή είτε λόγω ισοπαλίας.**
6. **Η μέθοδος getCurrentPlayer() επιστρέφει τον τρέχοντα παίκτη της σειράς.**
7. **Η μέθοδος setFold() αυξάνει το fold κατά ένα (fold++) ή θέτει το fold σε 0 αν τελειώσει ένας γύρος, και παρέχει τη σειρά στον επόμενο παίκτη.**
8. **Η μέθοδος makeChanges(ArrayList<Integer> cardsPosition) κάνει αλλαγές μετά από έναν ξεκινημένο γύρο με βάση τις παρεχόμενες θέσεις των καρτών.**

## Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο "View" στην υλοποίηση του παιχνιδιού "Sorry" σε Java είναι υπεύθυνο για το γραφικό περιβάλλον χρήστη (GUI) του παιχνιδιού και περιλαμβάνει κλάσεις όπως η MenuDialog, η WishDialog και η GraphicUI.

1. **Η κλάση MenuDialog:**
   * Αντιπροσωπεύει ένα διάλογο μενού για την αλληλεπίδραση με τον χρήστη, επιτρέποντας του να κάνει επιλογές από ένα μενού.
   * Διαθέτει έναν κατασκευαστή που δημιουργεί μια νέα οντότητα της MenuDialog και μια μέθοδο με το όνομα choice() που εμφανίζει το μενού και ανακτά την επιλογή του χρήστη.
2. **Η κλάση WishDialog:**
   * Αντιπροσωπεύει έναν διάλογο για την καταχώρηση ευχών ή επιλογών από τον χρήστη.
   * Παρόμοια με την MenuDialog, έχει έναν κατασκευαστή που δημιουργεί μια νέα οντότητα της WishDialog και μια μέθοδο με το όνομα choice() που εμφανίζει τις επιλογές ευχών και ανακτά την επιλογή του χρήστη.
3. **Η κλάση GraphicUI:**
   * Είναι υπεύθυνη για το γραφικό περιβάλλον χρήστη του παιχνιδιού και συνεργάζεται με την κλάση Controller.
   * Διαθέτει έναν κατασκευαστή που δημιουργεί ένα νέο παράθυρο και αρχικοποιεί κάποια κουμπιά και πάνελ για την έναρξη ενός νέου παιχνιδιού.
   * Διαθέτει επίσης μεθόδους όπως initComponents(), initButtons() και initBoard() που αρχικοποιούν τα στοιχεία, τα κουμπιά και το ταμπλό του παιχνιδιού αντίστοιχα.
   * Επιπλέον, περιλαμβάνει μια προσαρμοσμένη κλάση desktop pane για το σχεδιασμό του φόντου και μια κλάση ακροατή για την χειρισμό δράσεων που σχετίζονται με τις κάρτες.

## Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Στο διάγραμμα UML που παρέχεται, παρουσιάζονται οι κλάσεις και οι σχέσεις μεταξύ τους για το παιχνίδι "Sorry". Οι κλάσεις είναι οργανωμένες σε πακέτα όπως Model, View, και Controller, και κάθε κλάση περιέχει τα αντίστοιχα χαρακτηριστικά (attributes) και μεθόδους (methods).Για παράδειγμα, η κλάση Player στο πακέτο Model.Player έχει σχέσεις με τις κλάσεις Pawn και Turn, δείχνοντας ότι ο παίκτης διαθέτει πιόνια και συνδέεται με τη σειρά του παιχνιδιού. Η κλάση Board στο πακέτο Model αναπαριστά την ταμπλό και συνδέεται με τις κλάσεις τετραγώνων όπως HomeSquare, SafetyZoneSquare, και άλλες, δείχνοντας την οργάνωση της ταμπλός.Η κλάση Controller στο πακέτο Controller έχει μεθόδους όπως initializeGameElements() και movePawn(), και συνδέεται με τις κλάσεις Player και Board, δείχνοντας ότι διαχειρίζεται τη λογική του παιχνιδιού και την αλληλεπίδραση μεταξύ των παικτών και της ταμπλός.Στο πακέτο View, η κλάση GraphicUI αναπαριστά το γραφικό περιβάλλον και συνδέεται με την κλάση Controller, δείχνοντας την αλληλεπίδραση μεταξύ της γραφικής διεπαφής και της λογικής του παιχνιδιού.

## Λειτουργικότητα (Β Φάση)

Εχω κανει τα πολυ βασικα στις υλοποιησεις του project.

Επίσης, επειδή δεν ήμουν σίγουρος πώς ακριβώς να κάνω τη διαδικασία να τραβάει κάποιος κάρτα ή πώς να κινεί τα πιόνια, πρόσθεσα κάποια κουμπιά για να γίνει πιο εύκολο. Έτσι, μπόρεσα να κάνω τις κινήσεις στο παιχνίδι. Ελπίζω να μην είναι μεγάλο πρόβλημα αυτή η προσθήκη.

**Έναρξη Παιχνιδιού:** Φτιάχνω τους παίκτες και το ταμπλό για να αρχίσει το παιχνίδι.

Επιπροσθετα :

1. **Τράπουλα Καρτών:** Διαχειρίζομαι τις κάρτες και τις αφαιρώ από την τράπουλα. Υπάρχει ένα μικρό λάθος εδώ, όπου μου περισσεύουν 4 κάρτες.
2. **Κινήσεις με Κάρτες:** Μπορώ να κάνω τις βασικές κινήσεις με τις κάρτες 1, 2, και 4. Οι άλλες κάρτες δεν δουλεύουν ακόμη όπως πρέπει.
3. **Τσουλήθρες:** Αυτό το μέρος δουλεύει καλά.
4. **Fold:** Έχω κάνει να λειτουργεί και αυτό.
5. **Σειρά Παικτών:** Κρατάω σωστά τη σειρά των παικτών.
6. **Πιόνια στην Ίδια Θέση:** Διαχειρίζομαι τα πιόνια όταν είναι στην ίδια θέση.
7. **Επιλογές Παιχνιδιού:** Έχω φτιάξει επιλογές για να ξεκινήσει νέο παιχνίδι, να αποθηκεύσει και να φορτώσει παιχνίδι, και για έξοδο

## Συμπεράσματα

Το project ήταν απαιτητικό, αλλά με συνέπεια και πολλή δουλειά κάποιος μπορούσε να το ολοκληρώσει. Εγώ αντιμετώπισα δυσκολίες λόγω άλλων εργασιών σε διάφορα μαθήματα, που μου περιόρισαν τον διαθέσιμο χρόνο. Επίσης, στην αρχή χρειάστηκε χρόνος για να καταλάβω πώς να προχωρήσω με το project, γεγονός που με καθυστέρησε.

Παρατήρησα ότι πολλές από τις συναρτήσεις που είχα δημιουργήσει στην πρώτη φάση δεν τις χρησιμοποίησα τελικά και πρόσθεσα άλλες νέες. Αν και δεν έχω διαγράψει τις παλιές συναρτήσεις, συνολικά το project ήταν μια ευχάριστη εμπειρία και μου πρόσφερε την ευκαιρία να μάθω πολλά νέα πράγματα.