



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2023-2024

[ΤΙΤΛΟΣ ΑΝΑΦΟΡΑΣ]

*Εισαγωγή*

[Γιώργος Λιτσαρδάκης]

[5134]

[12/9/2023]

## Περιεχόμενα

1. Εισαγωγή.....	2
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model .....	2
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller.....	2
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	3
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....	3
6. Λειτουργικότητα (B Φάση).....	5
7. Συμπεράσματα .....	5

## 1. Εισαγωγή

Το μοντέλο αρχιτεκτονικής MVC (Model-View-Controller) είναι ένα πρότυπο σχεδίασης που χωρίζει μια εφαρμογή σε τρεις βασικές συνιστώσες: το model(που περιεχει όλες τις κλάσεις με τις λειτουργίες του παιχνιδιού), το view(που περιεχει τις αλληλεπιδράσεις του παίχτη με την εφαρμογή) και το controller(που αναλογα με τις αλληλεπιδράσεις του παίχτη, διαμορφώνει και το περιεχόμενο του παιχνιδιού).Παρακάτω υπάρχουν περισσότερες λεπτομερείς όσο αφορά την σχεδίαση του MVC και τα διαγράμματα UML.

## 2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Αρχικά δημιούργησα την abstract κλάση Square που αποτελεί ένα κουτάκι πάνω στο επιτραπέζιο ταμπλό. Κάθε ένα από τα στιγμιότυπα αυτής της κλάσης θα είχε μια ιδιαίτερη θέση πάνω στο ταμπλό. Έπειτα ακολουθούν οι υποκλίσεις της οι οποίες είναι η StartSlideSquare, InternalSlideSquare, EndSlideSquare, SafetyZoneSquare και StartSquare, όπου η κάθε μια έχει ξεχωριστό ρολό πάνω στο ταμπλό. Έπειτα έχουμε την abstract κλάση Card η οποία είναι υπεύθυνη για την γενίκευση του τραπουλόχαρτου του παιχνιδιού. Για να είμαστε περισσότερο ακριβείς έφτιαξα τις υποκλάσεις SorryCard και NumberCard οι οποίες διαχωρίζουν την κάρτα SORRY από τις υπόλοιπες καθώς δεν έχει έναν αριθμό πάνω σε αυτήν. Γνωρίζοντας ότι η NumberCard είναι εξίσου μια ευρεία έννοια, δημιούργησα τις υποκλάσεις SimpleNumberCard, η οποία αποτελεί τις κάρτες με αριθμό 3,5,8,12, αφού δεν έχουν κάποια σημαντική διαφορά η μια από την άλλη και μπορούν να θεωρηθούν ως "απλές" κάρτες, και τις υποκλάσεις NumberOneCard, NumberTwoCard, NumberFourCard, NumberSevenCard, NumberTenCard, και NumberElevenCard, αφού η κάθε μια από αυτές έχει και μια ιδιαιτερότητα. Επίσης χρησιμοποιώ μια κλάση Deck η οποία αποτελεί μια τράπουλα από τις κάρτες που ανέφερα πιο πάνω και τις αντίστοιχες λειτουργίες όπως να τραβηχτεί ένα τραπουλόχαρτο από αυτήν. Επιπρόσθετα έχω μια κλάση Slide η οποία είναι υπεύθυνη για την λειτουργία της τσουλίθρας του παιχνιδιού. Κάθε κλάση Slide περιεχει ένα Square τύπου StartSlideSquare, ένα Square τύπου EndSlideSquare και έναν πίνακα ο οποίος έχει μέσα squares τύπου InternalSlideSquare. Τέλος έχω τις βασικότερες κλάσεις Board, Pawn και Player οι οποίες είναι υπεύθυνες για τις λειτουργίες του ταμπλό, πιονιού και παίχτη αντίστοιχα.

## 3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Το πακέτο Controller περιεχει μια μόνο κλάση την Controller, η οποία είναι υπεύθυνη για την αρχικοποίηση και έναρξη του παιχνιδιού. Αναλυτικότερα σε αυτή την κλάση αρχικοποιώ το ταμπλό του παιχνιδιού, τους παίχτες, τα πιόνια του κάθε παίχτη ακόμα και την τράπουλα. Άλλη μια λειτουργία της κλάσης μου είναι η αλληλεπίδραση της εικόνας (το ταμπλό και τα πατήματα του κάθε παίχτη) με το ίδιο το παιχνίδι. Για παράδειγμα, μέσα σε αυτή την κλάση περιέχονται μέθοδοι οι οποίες είναι υπεύθυνες για τα κουμπιά του παιχνιδιού, τα οποία είναι το κουμπί "τράβηγμα κάρτας", Start\_game και fold. Επίσης εξαιτίας του χρονικού προβλήματος δεν κατάφερα να συνδυάσω το κουμπί fold με το σήκωμα κάρτας και να μπορεί

ο κάθε παίχτης να το χρησιμοποιεί υπό κάποιες συγκεκριμένες κάρτες και συνθήκες παιχνιδιού. Στο δικό μου τελικό παιχνίδι το κουμπί fold κάνει μια απλή λειτουργία, η οποία είναι να ακυρώνει την κάρτα και να "δίνει" την σειρά στον επόμενο παίχτη. Τέλος, για αυτήν την κλάση είχα το σκεπτικό να χρησιμοποιώ περισσότερο τους ελέγχους δυνατότητας παιξίματος της σηκωμένης κάρτας του κάθε πιονιού και της κάθε ανάλογης κάρτας.

#### 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View αποτελείται από μια κλάση. Αυτή η κλάση με όνομα View παρουσιάζει εικονικά το παιχνίδι, δηλαδή μέσω κουμπιών και διάφορα dialogs, ο παίχτης θα μπορεί να αλληλοεπιδρά με το παιχνίδι. Επίσης, αυτή η κλάση είναι υπεύθυνη για την ανανέωση του εικονικού περιβάλλοντος καθώς και κάποιους μικρό ελέγχους όπως αυτούς της τσουλήθρας ή ακόμα και την κίνηση των ποινών πάνω στο ταμπλό, έπειτα από κάποια εντολή του Controller. Κάποιες βασικές μέθοδοι που υπάρχουν μέσα στην κλάση είναι η `createstartpanel` η οποία είναι η πρώτη επαφή με τους παίχτες, με σκοπό την έναρξη του παιχνιδιού, η `creategamepanel`, η οποία φτιάχνει τα γραφικά του παιχνιδιού και η `updateinfobox`, η οποία ενημερώνει την εικόνα της σηκωμένης κάρτας. Έπειτα, άλλες 3 βασικές μεθόδους που υπάρχουν μέσα στην κλάση View είναι η `updateinfobox`, η `setAllTogether` και η `updateturn` οι οποίες ανανεώνουν το κουτί με τις πληροφορίες σχετικά με την κάθε κάρτα, συγχρονίζουν τα κουμπιά με το γραφικό περιβάλλον, και αλλάζουν την σειρά των παιχτών (με κατάλληλο μνήμα) αντίστοιχα. Τέλος, να σημειωθεί ότι υπάρχουν ακόμα κάποιες βοηθητικές μεθόδους οι οποίες χρησιμοποιούνται περισσότερο στην κλάση Controller.

#### 5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Η κληρονομικότητα μεταξύ των προσωρινών κλάσεων μου είναι η εξής:

Οι κλάσεις `SafetyZoneSquare`, `StartSquare`, `StartSlideSquare`, `InternalSlideSquare` και `EndSlideSquare` είναι υποκλάσεις της κλάσης `Square`.

Οι κλάσεις `NumberCard` και `SorryCard` είναι υποκλάσεις της κλάσης `Card`.

Οι κλάσεις `SimpleNumberCard`, `NumberOneCard`, `NumberTwoCard`, `NumberFourCard`, `NumberSevenCard`, `NumberTenCard` και `NumberElevenCard` είναι υποκλάσεις της κλάσης `NumberCard`.

Η απλή συσχέτιση και αλληλεπίδραση μεταξύ των κλάσεων μου είναι η εξής:

Οι κλάσεις `Board`, `Pawn`, `EndSlideSquare`, `InternalSlideSquare` και `StartSlideSquare` συσχετίζονται και έτσι συνδέονται με την κλάση `Slide`.

Οι κλάσεις `Player`, `Board` και `View` συσχετίζονται και έτσι συνδέονται στην κλάση `Controller`.

Οι κλάσεις `Pawn` και `Card` συνδέονται με την κλάση `Player`.

Οι κλάσεις `Pawn`, `Deck` και `Board` συνδέονται με την κλάση `View`.

Οι κλάσεις `Deck`, `Pawn` και `Square` συνδέονται με την κλάση `Board`.

Η κλάση `Card` συνδέεται με την κλάση `Deck`.

Παραπάνω αναλυτικότερα διαγράμματα UML έχω στο pdf που έστειλα.



## 6. Λειτουργικότητα (B Φάση)

Στην φάση B κατάφερα αρκετά πράγματα. Αρχικά , γίνεται σωστή και λεπτομερές αρχικοποίηση των παιχτών, του ταμπλό, της τράπουλας αλλά και των γραφικών. Επίσης, υπάρχει τήρηση της σειράς, δηλαδή έπειτα από το παίξιμο μιας κάρτας η του πατήματος του κουμπιού fold (ακόμα και την πιθανότητα το παίξιμο της κάρτας να είναι αδύνατο) η σειρά "πάει" στον επόμενο παίκτη. Επιπλέον, κατάφερα να υλοποιήσω τις κάρτες 1,2,3,5,8,12 και sorry (μαζί με τον έλεγχο για Safety zone). Αναλυτικότερα οι κάρτες αλληλοεπιδρούν στο παιχνίδι όπως ακριβώς θα έπρεπε, με το μοναδικό μικρό λάθος το γεγονός ότι ο έλεγχος γίνεται μέσα στην κλάση Controller και όχι στις ανάλογες μεθόδους isitplayable() που βρίσκονται μέσα στην κάθε κλάση της κάθε κάρτας. Επίσης γίνεται σωστή λειτουργία της κάθε τσουλήθρας με το μοναδικό ελάττωμα ότι ο έλεγχος και η κίνηση πραγματοποιείται στην αρχή της μεθόδου updatepawn που βρίσκεται μέσα στην κλάση View, και όχι στην ανάλογη κλάση που δημιούργησα στην πρώτη φάση με όνομα Slide. Αυτά τα 2 λαθάκια (που οι έλεγχοι γίνονται σε λάθος κλάση) έγιναν υποχρεωτικά λόγω της πίεσης του χρόνου. Τέλος , ο έλεγχος για το παίξιμο της κάθε κάρτας περιλαμβάνει και τον κανόνα για τα πιόνια στην ιδιά θέση , και ακόμα υπάρχει σωστή χρήση της θέσης HOME δηλαδή την θέση τερματισμού(ακόμα και την απαίτηση για ακριβή αριθμό για τερματισμό). Τα πράγματα που δεν κατάφερα να υλοποιήσω είναι JUnit Tests και τις κάρτες 4,10,7,11. Γενικότερα το μόνο θέμα που πιστεύω ότι υπάρχει στον κώδικα μου πέρα από αυτά είναι ότι δεν χρησιμοποιώ πλήρως τις μεθόδους των υποκλίσεων(π.χ. τις μεθόδους για έλεγχο παιχνιδιματος που υπάρχουν μέσα στην κάθε ανάλογη κλάση της κάθε κάρτας).

## 7. Συμπεράσματα