



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2022-2023

STRATEGO

Εισαγωγή

Λέσι Σκελζέν

4486

25/11/2022

Περιεχόμενα

1. Εισαγωγή	2
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model	2
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	6
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller	6
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML	7
6. Λειτουργικότητα (B Φάση).....	8
7. Συμπεράσματα	8

1. Εισαγωγή

Το μοντέλο που θα χρησιμοποιήσουμε στην εργασία μας είναι το MVC(Model, View, Controller), το οποίο θα αναλύσουμε στις επόμενες ενότητες της αναφοράς μας.

Αρχικά εξηγούμε τις κλάσεις και τις μεθόδους που χρησιμοποιήσαμε στο πακέτο Model, το οποίο χρησιμοποιείται για την αποθήκευση των δεδομένων της εργασίας μας.

Επτά, συνεχίζουμε με το πακέτο View, στο οποίο υλοποιούμε κώδικα που σχετίζεται με τα γραφικά της εργασίας μας.

Τέλος, έχουμε το πακέτο Controller στο οποίο χρησιμοποιείται να ενώσει το Model με το View μαζί, δηλαδή να παίρνει δεδομένα από το Model και να τα αποτυπώνει στο View, καθώς και το αντίθετο αφού η οποιαδήποτε κίνηση πάνω στο View μπορεί να αλλάξει τα δεδομένα μας, κανοντας έτσι αυτοματη ενημέρωση χάρη στον Controller.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Το πακέτο Model είναι χωρισμένο σε τρία πακέτα:

1. Dashboard το οποίο περιέχει μέσα την κλάση Board η οποία αποθηκεύει πληροφορίες για το κομμάτι του παιχνιδιού πχ, στατιστικά, συνολικοί γυροί κλπ.

Οι μέθοδοι που υλοποιούνται μέσα σε αυτήν την κλάση είναι οι εξής:

- a) Board(), η οποία είναι ο constructor της κλάσης αυτής στην οποία θα γίνουν κάποιες αρχικοποιήσεις με το ξεκίνημα το παιχνιδιού.
- b) SetTotalCaptures() η οποία θα αθροίζει συν 1 κάθε φορά που θα γίνεται μια αιχμαλωτισή στο παιχνίδι ώστε να κρατάει το συνολικό αθροισμα. Επίσης υπάρχει και η GetTotalCaptures για να μπορούμε να

διαβαζουμε αυτό το αθροισμα οσο ειμαστε εκτος κλασης αφου η μεταβλιτη αυτη είναι ιδιοτικη.

c) Εχουμε επισης δυο συναρτησεις Set και Get για την ιδιοτικη μεταβλιτη Player1Turn την οποια θα χρησιμοποιουμε για να αναθετουμε στον Player1Turn True σε περιπτωση που εχει σειρα ο Player1 αλλιως false στην περιπτωση που εχει σειρα ο Player2.

d) Στη συνεχεια εχουμε τον setter και getter της ιδιωτικής μεταβλιτης lessArmy στην οποια δηλωνουμε αν ο παικτης επιθυμει λιγοτερο στρατο η όχι στο ξεκινημα της παρτηδας.

e) Όπως και στο πρωιγουμε ετσι εχουμε setter και getter για την μεταβλιτη neverGoBack οπου λεμε αν ο παικτης θελει τα πονια να μην μπορουν να υποχωρισουν η όχι.

f) Άλλο ένα ζευγαρι set και get που εχουμε είναι αυτό για την τιμη totalRounds που μας λεει ποσοι γυροι εχουν παιχτει συνολικα.

g) Τελος εχουμε τους setter και getter της μεταβλιτης captures η οποια είναι ενας πεινακας από τις συνολικες αιχμαλωτησεις που γινονται και από τους δυο παικτες.

h) Για την Β φαση προσθεσαμε ένα setter και getter για την μεταβλιτη totalrounds η οποια θα αποθηκευει τους συνολικους γυρους που παιζονται στο παιχνιδι.

2. Pawns Το οποιο περιεχει πληροφοριες σχετικα με τα πονια του παιχνιδιου. Το πακετο αυτό χωριζεται σε:

a) Μια abstract class Piece η οποια κανει extend JButton το οποιο θα μας χρειασται στο κομματι της υλοποιησης

κάθως το κάθε στιγμοτυπο αυτης της κλασης θα είναι ένα κουμπι.

Οι μεθοδοι που υλοποιει αυτή η κλαση είναι οι εξεις:

- a. Ένας constructor Piece στον οποιο θα γινονται αρχικοποιησεις οσο αφορα τα πιονια της κάθε ομάδας.
- b. Μια abstract void attack() η οποια θα χρησιμοποιηται από τις υποκλασεις της για να γινεται μια επιθεση μεσα στο παιχνιδι.
- c. Στη συνεχεια εχουμε ένα ζευγαρι από setter και getter της μεταβλητης Alive οπου θα αποθηκευουμε αν το κάθε πιονι είναι ζωντανό η όχι.
- d. Μετα εχουμε setter και getter για την μεταβλητη Place η οποια μας δινει την θεση του κάθε πιονιου.
- e. Συνεχιζουμε με άλλο ένα ζευγος από setter και getter της μεταβλητης power η οποια μας δινει την δυναμη του κάθε πιονιου.
- f. Τελος εχουμε setter και getter για την μεταβλητη color οπου θα λεμε αν το χρωμα του πιονιου θα είναι μπλε η κοκκινο.
- g. Για τη B φαση εχουμε προσθεση ένα setter και getter οπου επαιξεργαζονται μια μεταβλητη τυπου Icon στην οποια αποθηκευουμε την φωτογραφια που βαζουμε πανω σε κάθε κουμπι.
- h. Άλλο ένα ζευγος setter και getter για την κλαση αυτή είναι της μεταβλητης hasSaved η οποια μας λεει αν ένα κουμπι εχει ξαναδιασωσει, αν ναι δεν μπορεί ξανα.
- i. Επισης εχουμε ένα setter και getter της chosenByRandom η οποια χρησιμοποιηται για την τυχαια αρχικοποιηση των παιχτων της κάθε ομάδας (Μπλε και κοκκινοι).

- j. Τέλος έχουμε setter και getter για την μεταβλητή `canMoveThere` την οποία χρησιμοποιούμε στην `action performed` ώστε να δούμε τι συμπεριφορά πρέπει να έχει το κάθε κλικ σε ένα κουμπι καθώς κάθε κλικ μπορεί να κάνει διαφορετικά πράγματα.
 - b) Την υποκλάση `MoveablePiece` η οποία κάνει `extend` την `Piece` και περιέχει τις εξής μεθόδους:
 - a. Έναν `Constructor MoveablePiece` όπου γίνονται οι καταληλές αρχικοποιήσεις για τα πιόνια που κινούνται πάνω στο ταμπλο.
 - c) Την υποκλάση `ImmoveablePiece` η οποία κάνει και αυτή `extend` την υπερκλάση `Piece` και έχει τις εξής μεθόδους:
 - a. Έναν `constructor ImmoveablePiece` όπου γίνονται οι καταληλές αρχικοποιήσεις για τα πιόνια που δεν κινούνται(παγίδες και σημαία).
3. Players στο οποίο αποθηκεύουμε πληροφορίες σχετικά με τους παίκτες του παιχνιδιού. Το πακέτο αυτό περιέχει την κλάση `Player` με τις εξής μεθόδους:
- a. Έναν `Constructor Player` όπου γίνονται οι καταληλές αρχικοποιήσεις για τους παίκτες του παιχνιδιού.
 - b. setter και getter για το όνομα του κάθε παίκτη.
 - c. setter και getter για τις αιχμαλώτησεις του κάθε παίκτη.
 - d. setter και getter για το ποσοστό επιτυχίας στις νηκες του `attack` κάθε παίκτη χωριστά.
 - e. setter και getter για τις συνολικές διασώσεις του κάθε παίκτη καθώς επιτρέπονται μερι δυο.
 - f. Στη Β φάση αυτής της κλάσης έχουμε προσθέσει ένα setter και getter για συνολικαίς επιθεσίς του κάθε παίκτη το οποίο χρησιμοποιούμε για να βγάλουμε το συνολικό ποσοστό επιτυχίας.

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το πακέτο View που όπως ειπαμε στην εισαγωγή περιέχει τα γραφικά της εργασίας μας, έχει τις εξής κλάσεις:

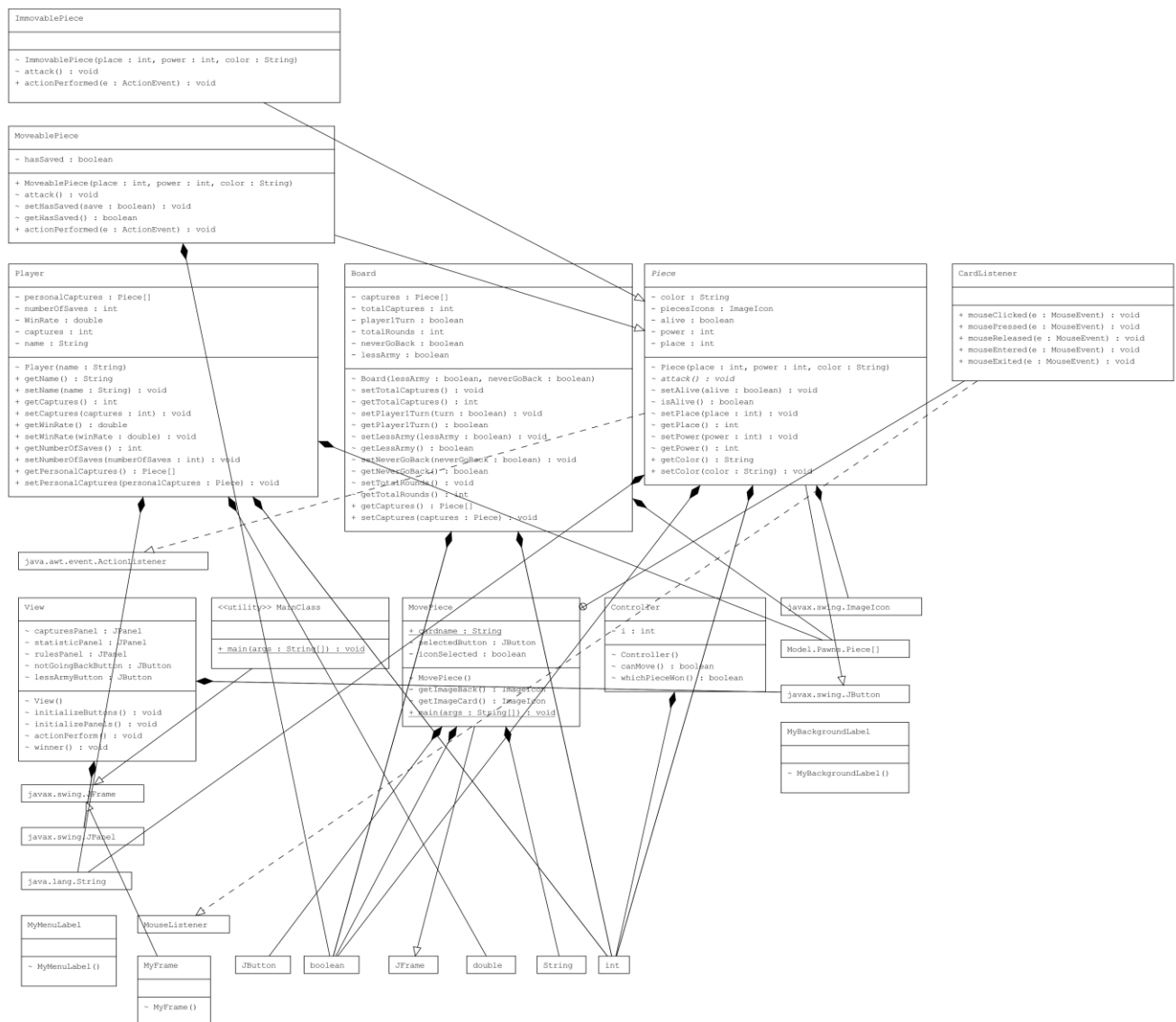
- a) MyFrame στην οποία θα έχουμε πληροφορίες για το frame μας, πχ(μεγεθος, visibility κλπ.).
- b) MyBackgroundLabel όπου θα έχουμε πληροφορίες για το label όπου θα βαζουμε τα buttons μας.
- c) MyMenuLabel όπου θα έχουμε πληροφορίες για το label του μενου μας, εκει δηλαδη που θα λεει ποιος παικτης εχει σειρα κλπ.
- d) View όπου θα γινονται διαφορες αρχικοποιησεις για τα κουμπια, labels κλπ.

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Το πακέτο controller περιεχει μια κλαση, την controller. Η κλαση αυτή περιεχει τις εξης μεθοδους:

- a) Τον Constructor controller οπου θα γινονται αρχικοποιησεις σχετικα με το παιχνιδι μας.
- b) Την μεθοδο canMove() οπου επιστρεφει το αν μπορεί ένα πονει να κανει μια συγκεκριμενη κινιση η όχι.
- c) Τελος την μεθοδο WichPieceWon η οποια μεσα σε μια διαμαχη θα μας επιστρεφει πιο πioni νικησε.

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



6. Λειτουργικότητα (Β Φάση)

- Τήρηση σειράς και τοποθέτηση πονιών (10%) Λειτουργεί κανονικά.
- Σωστή Κίνηση πονιών και ανιχνευτή (12%) Λειτουργεί κανονικά,
- Επίθεση πονιών (10%) Λειτουργεί κανονικά.

- Επίθεση εξολοθρευτή-δράκου (7%) Λειτουργει κανονικα.
- Εξουδετέρωση παγίδας από νάνο (5%) Λειτουργει κανονικα.
- Διάσωση πιονιού που βγήκε από το παιχνίδι (10%) Δεν λειτουργει, απλα εμφανιζει ποια πονια είναι αιχμαλωτισμενα αλλα δεν μπορείς να διασώσεις κανένα.
- Τέλος παιχνιδιού-κατάκτηση σημαίας (6%) Λειτουργει κανονικα.
- Μενού πληροφοριών στο πλάι (12%) Υπολειτουργει,υπαρχει επικοινωνια view model μεταξυ controller απλα δεν εχει το κομματι του να εμφανιζονται τα αιχμαλωτισμενα πονια και δεν λειτουργουν οι ενεργοι κανονες
- Game mode: Μειωμένος Στρατός (8%) Δεν λειτουργει
- Game mode: Καμία Υποχώρηση (Bonus 5%) Δεν λειτουργει

7. Συμπεράσματα

Στο παραδοτέο έχω συμπεριλαβει ανα αρχειο με UML διαγραμματα τα οποια έχω κανει αυτοματο generate και λοιπουν καποιες διασυνδεσεις. Είναι ένα ωραιο Project που απετουσε σχετικα λιγη ωρα απασχολισης.