



**Τεχνολογία Λογισμικού
Εργασία 3η
Ακαδημαϊκό έτος 2018-2019**

Ζητούμενο 1 : Αλλαγές στις περιπτώσεις χρήσης

1.Τίτλος περίπτωσης χρήσης: Επιλογή ρυθμίσεων

1. παίκτης επιλέγει το πλήκτρο "New Game" στην Οθόνη έναρξης του παιχνιδιού.
2. Το σύστημα εμφανίζει την δεύτερη οθόνη Δεύτερη των ρυθμίσεων.
3. Ο χρήστης επιλέγει τον αριθμό και το χρώμα των παικτών.
4. Ο χρήστης πατώντας "Enter" δηλώνει την έναρξη του παιχνιδιού και το παιχνίδι αρχίζει.

1.1 Εναλλακτικές ροές

1.2.1. Εναλλακτική ροή 1

- 1α. Ο χρήστης πατάει το κουμπί «Load Game».
- 1β. Φορτώνεται ένα παλιότερο παιχνίδι που δεν είχε τερματιστεί και ο χρήστης μπορεί να το συνεχίσει.
- 1γ. Εμφανίζεται η οθόνη του ταμπλό και ο παίκτης συνεχίζει το παιχνίδι του.

1.2.2. Εναλλακτική ροή 2

- 1α. Ο χρήστης πατάει το κουμπί «Exit».
- 1β. Τερματίζει η εφαρμογή και ο χρήστης βγαίνει από αυτήν.

2.Τίτλος περίπτωσης χρήσης: Ενέργεια παίκτη

Σύντομη περιγραφή: Ο χρήστης ρίχνει τα ζάρια, μεταφέρεται το πιόνι του στο αντίστοιχο τετράγωνο και εκτελεί τις ανάλογες ενέργειες.

Βασική Ροή

1. Ο χρήστης ρίχνει το ζάρι.
2. Μεταφέρεται το πιόνι του παίκτη στο αντίστοιχο τετράγωνο του ταμπλό.
3. Ελέγχεται αν η ιδιοκτησία που σταμάτησε ο παίκτης έχει αγοραστεί.
4. Αν δεν έχει αγοραστεί και ο παίκτης επιλέξει να την αγοράσει.
5. Γίνεται έλεγχος για το αν το υπόλοιπο του παίκτη επαρκεί για την αγορά αλλιώς η ενέργεια ακυρώνεται εμφανίζοντας κατάλληλο μήνυμα.
6. Αφαιρείται το αντίστοιχο υπόλοιπο από το λογαριασμό του.
7. Τοποθετείται το χρώμα του παίκτη στο τετράγωνο ιδιοκτησίας που αγόρασε.
8. Παίζει ο επόμενος παίκτης.

2.2. Εναλλακτικές ροές

2.2.1. Εναλλακτική ροή 1

1. α. Ο χρήστης δεν εκτελεί κάποια ενέργεια και πατάει το πλήκτρο "Enter".
2. β. Πάμε στο Βήμα 8.

2.2.2. Εναλλακτική ροή 2

1α. Ο χρήστης πατάει το κουμπί «Έξοδος».

1β. Η εφαρμογή τερματίζει.

2.2.3. Εναλλακτική ροή 3

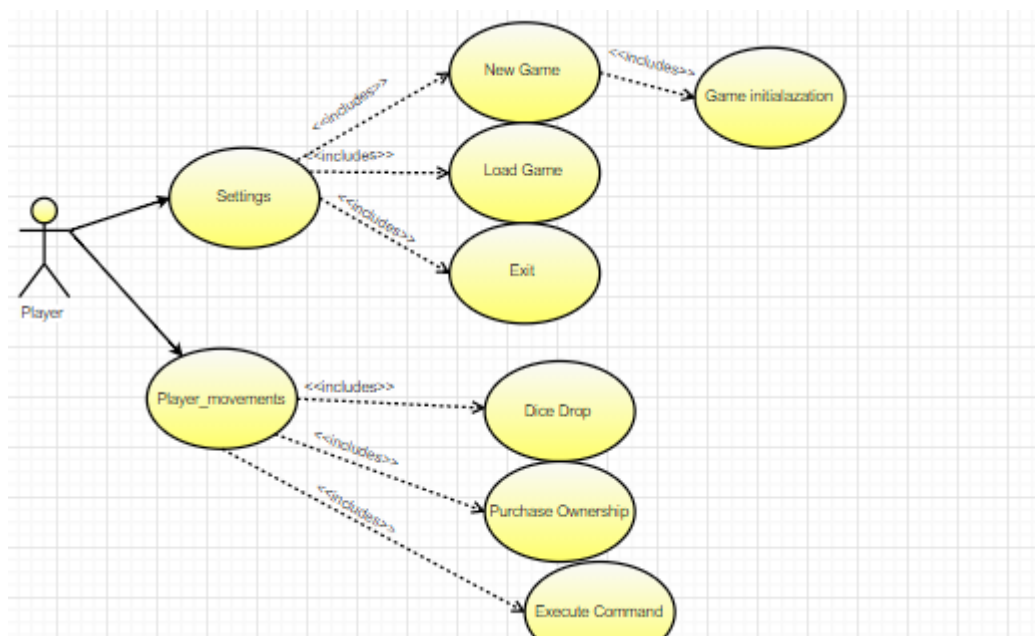
4α. Εμφανίζεται μήνυμα ότι την ιδιοκτησία αυτή την έχει ήδη αγοράσει και ο χρήστης πατάει το κουμπί "Enter" για να συνεχιστεί το παιχνίδι.

4β. Πάμε στο βήμα 8.

Εναλλακτική ροή 5

3α. Εμφανίζεται η κάρτα εντολής και πραγματοποιούνται αυτόματα οι ενέργειες που απαιτούνται.

3β. Πάμε στο βήμα 8.



Ζητούμενο 2:Domain Model

Τα ουσιαστικά που εντοπίστηκαν από το αρχικό κείμενο παρτίθενται παρακάτω:

Λίστα ουσιαστικών

οθόνη	σύστημα
Χρήστης	εντολή
παιχνίδι	ιδιοκτησία
παίκτης	ενέργειες
χρώμα	ενοίκιο
πλήθος	ιδιοκτήτης
ταμπλό	τετράγωνο
εντολές	
χρηματικό υπόλοιπο	
ζάρι	

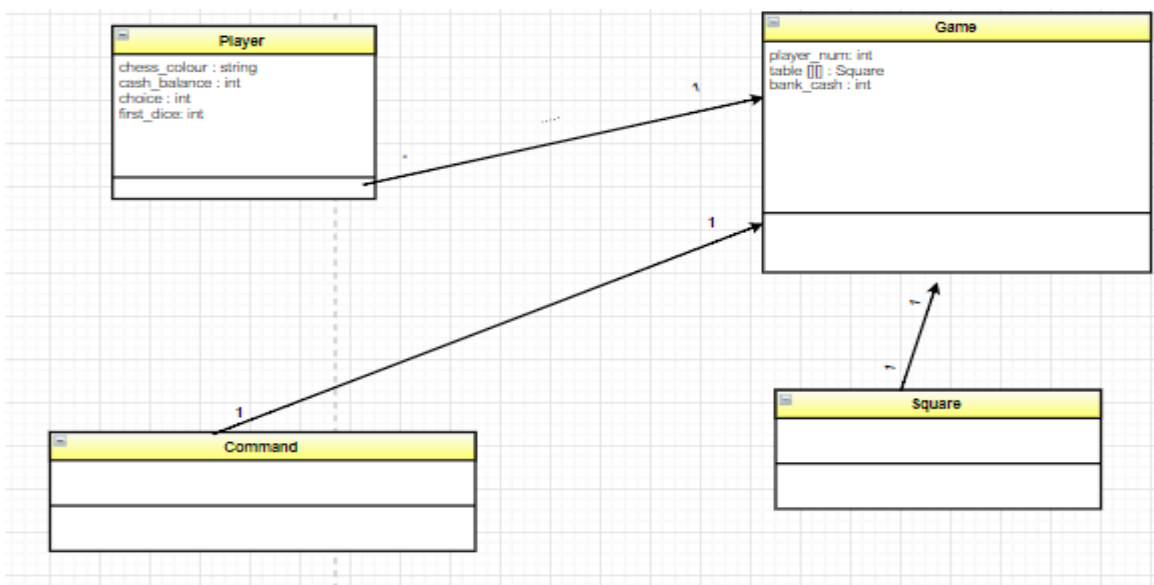
Περιορισμός υποψηφίων κλάσεων

Περιορισμός της λίστας απαλοίζοντας:

1. Αναφορές στο ίδιο το σύστημα λογισμικού που αναπτύσσουμε : (Σύστημα,οθόνη)
2. Αναφορές σε χειριστές του συστήματος που πρόκειται να αναπτύξουμε καθώς βρίσκονται "έξω" από τα όρια του συστήματος (νικητής,ιδιοκτήτης)
3. Αναφορές σε οντότητες που βρίσκονται εκτός του πεδίου του προβλήματος (χρήστης).
4. Ουσιαστικά που πιθανόν να αποτελέσουν ιδιότητες άλλων κλάσεων (χρώμα,πλήθος,ταμπλό,χρηματικό υπόλοιπο,ζάρι,ενέργεια,ιδιοκτησία,εντολή,ενοίκιο).

Υποψήφιες κλάσεις
Παίκτης
Τετράγωνο
Εντολή
Παιχνίδι

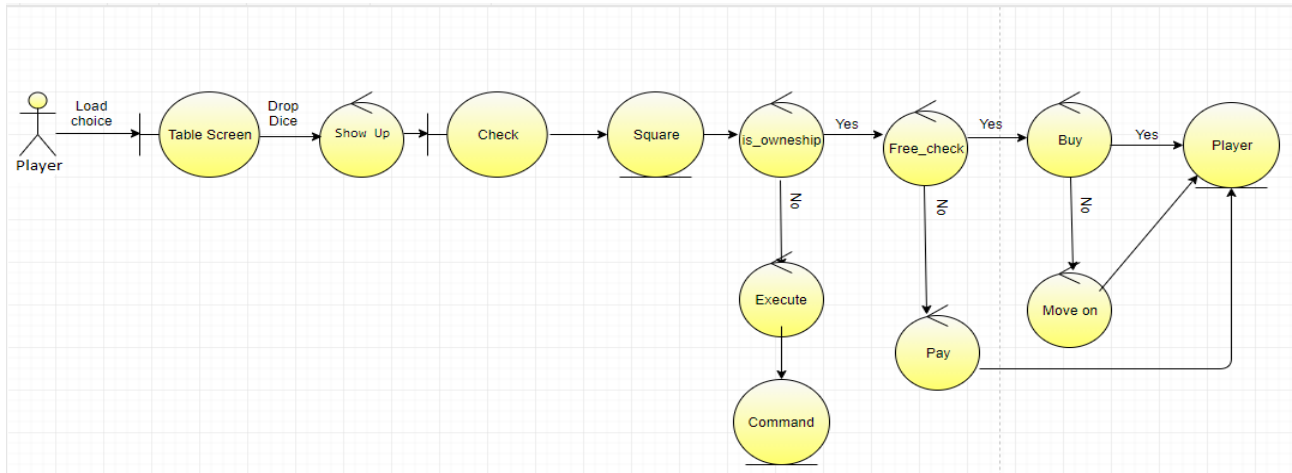
Αρχικό Domain Model



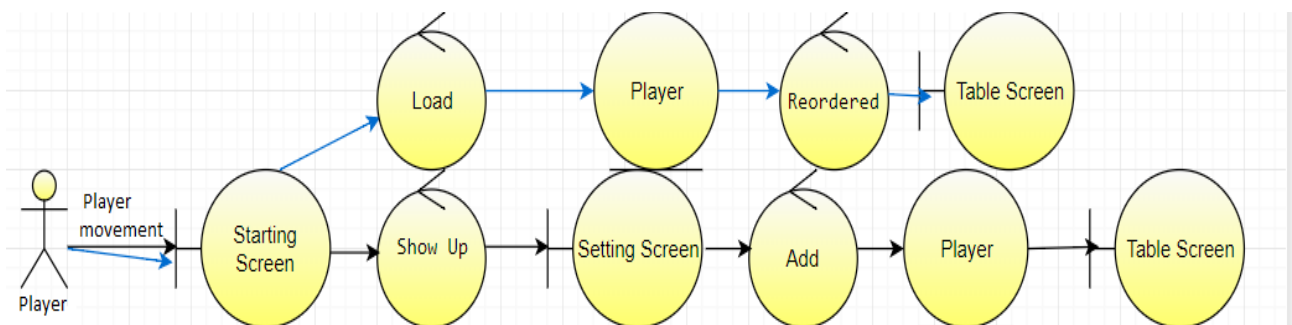
Ζητούμενο 3

Διαγράμματα ευρωστίας (robustness diagrams)

α) Για το use case **Επιλογή ρυθμίσεων**:



β) Για το use case **Ενέργεια παίκτη**:



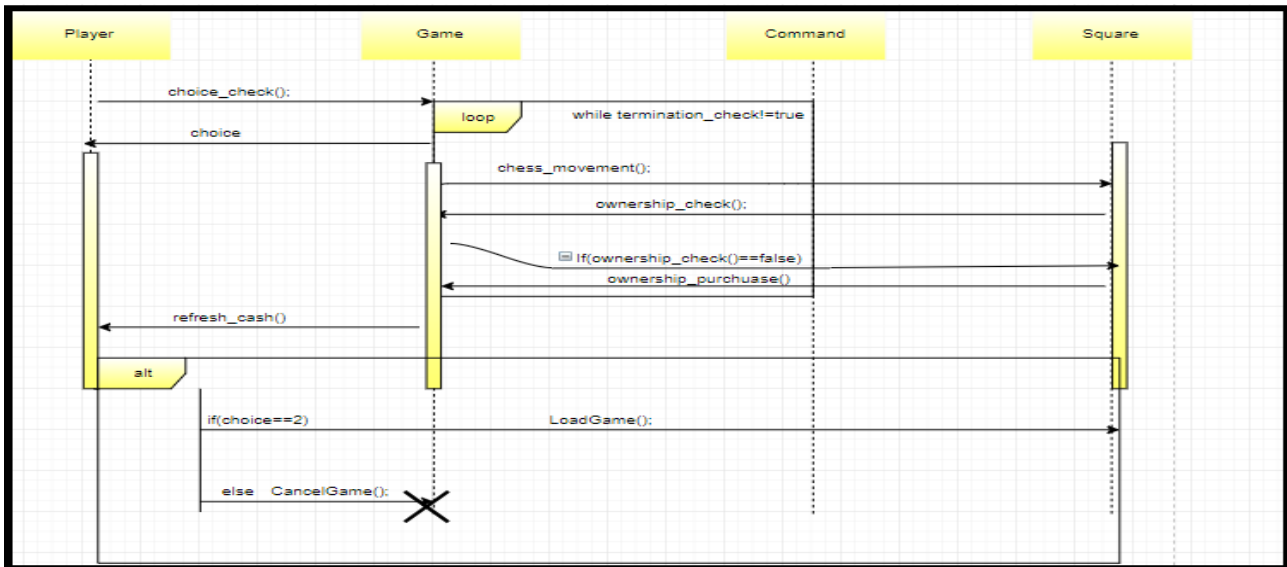
Στο παραπάνω διάγραμμα σημειώνεται με:



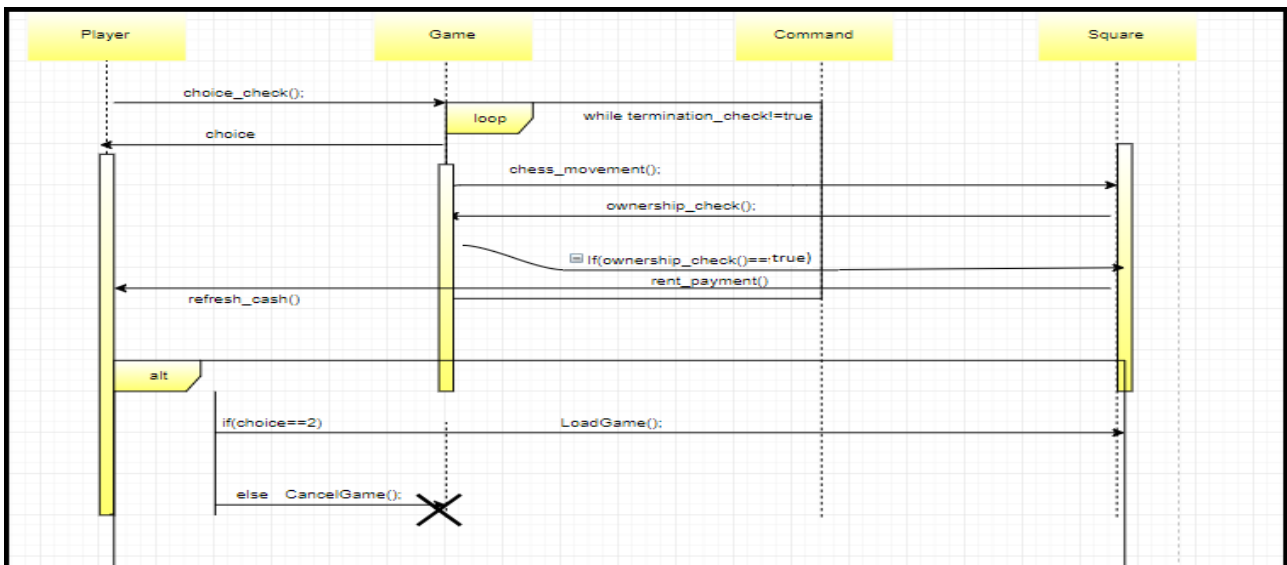
Ζητούμενο 4

Διαγράμματα ακολουθίας (sequence diagrams)

α) Αγορά ιδιοκτησίας

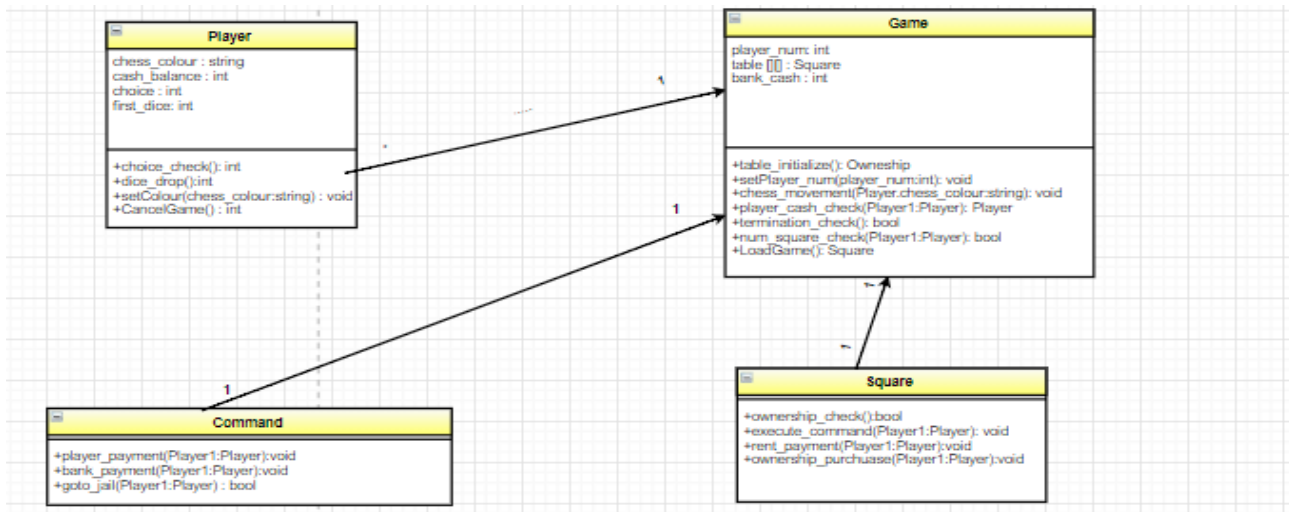


β) Πληρωμή ενοικίου



Ζητούμενο 5

Τελικό domain model



-Για να μην γίνει χρήση μιας επιπλέον κλάσης για τις ιδιοκτησίες δημιουργήθηκε ένας πίνακας table [][] ο οποίος είναι τύπου square με την λογική ότι ένα τετράγωνο μπορεί να είναι είτε εντολή είτε ιδιοκτησίας.

-Η κλάση Game θα περιέχει τη main μας δηλαδή θα είναι η κεντρική κλάση τους συστήματος μας και θα είναι αυτή που θα παίζει τον κύριο ρόλο για την ενημέρωση του συστήματος μας κάνοντας χρήση των συναρτήσεων που έχει μέχρι η συνάρτηση termination_check() να επιστρέψει true και να τερματίσει το παιχνίδι.

Ζητούμενο 7

Τα βασικά δομικά στοιχεία του κώδικα σε java παρατίθενται παρακάτω:

```

import java.util.Scanner;
import java.util.*;
public class Player extends Game {
Player:Player()
{          /*void constructor*/}
private string chess_colour;
int cash_balance;
int choice;
int first_dice;

public int choice_check()
{
Scanner scanner = new Scanner (System.in);
System.out.println("1.New Game\n2.Load Game\n3.Exit\nPlease Enter a choice:");
choice = scanner.nextInt(); // Get what the user types.
while(choice<1 || choice>3)
{
System.out.println("Wrong choice!\nPlease try again.");
System.out.println("1.New Game\n2.Load Game\n3.Exit\nPlease Enter a choice:");
System.out.println("1.New Game\n2.Load Game\n3.Exit\nPlease Enter a choice:");
choice = scanner.nextInt(); // Get what the user types.
}

if(choice==1)
{
Game.table_initialize();
first_dice=dice_drop();
}else if(choice==2)
{
Game.LoadGame();
}else
{
System.exit(0);
}
int dice_drop()
{
int min=1;
int max=6;
Random rand = new Random();
// nextInt as provided by Random is exclusive of the top value so you need to add 1
int randomNum = rand.nextInt((max - min) + 1) + min;
return randomNum;
}
public void setColour(String chess_colour) {
this.chess_colour = chess_colour;
}

}

```

Παρατήρηση:Για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκε το draw.io