

**Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών**

**ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός**

**Διδάσκων: Ι. Τζίτζικας**

**Χειμερινό Εξάμηνο 2023-24**

**Project**

**Βολονάκης Μάριος CSD5251**

*Εισαγωγή*

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

**Περιεχόμενα**

**1.** **Εισαγωγή 1**

**2.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου Model 1**

**3.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller 1**

**4.** **Η Σχεδίαση και οι Κλάσεις του Πακέτου View 2**

**5.** **Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML 2**

**6.** **Λειτουργικότητα (Β Φάση) 2**

**7.** **Συμπεράσματα 2**

* Εισαγωγή

Εδώ θα περιγράψετε σε γενικές γραμμές ποιο μοντέλο χρησιμοποιήσατε για την εργασία σας (MVC) και θα αναφέρετε πολύ συνοπτικά τι περιέχουν οι υπόλοιπες ενότητες της αναφοράς.

* Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Εδώ θα περιγράψετε τo σχέδιο υλοποίησης της προγραμματιστικής εργασίας του πακέτου model. Συγκεκριμένα. στην Α φάση του Project θα περιγράψετε τη σχεδίαση κάθε κλάσης (π.χ., αν χωρίσατε μία κλάση σε υποκλάσεις και για ποιους λόγους), ποιες μεθόδους σκοπεύετε να χρησιμοποιήσετε στη Β φάση του Project (και για ποιο λόγο είναι χρήσιμες, ποια είναι η λειτουργικότητα τους). Μπορείτε να το χωρίσετε και σε υπό ενότητες την ενότητα αυτή (πχ ενότητα για την κλάση και τις υποκλάσεις της Tile, ενότητα για την κλάση Player κλπ.). Επιπλέον, μπορείτε να συμπεριλάβετε διαγράμματα UML για να σας βοηθήσουν στην επεξήγηση των κλάσεων. Στη Β φάση του project, συμπληρώνετε τις λεπτομέρειες της υλοποίησης (πχ αλγόριθμοι που χρησιμοποιήθηκαν) και τυχόν αλλαγές.

public class Deck οπου θα περιεχονται:

public Deck() //αρχικοποιηση καρτων/deck αναλογα με τους παιχτες

public Card get(int players)

public int D\_size()

public void remove (int i)

Public class Player για την αρχικοποιηση παιχτων.οπου θα περιεχονται:

private ArrayList<Pawn> pawns=new ArrayList<Pawn>()

public Player (int color)

public ArrayList<Pawn> getPawns

public abstract class Square οπου θα περιεχονται:

private int color

private boolean full

public void setColor (int c)

public int getColor()

public void setFull (boolean f)

public boolean getTaken ()

public class SlideSquare extends Square

//οπου int c παρακατω αρχικοποιήται το χρωμα του τετραγώνου.

public SlideSquare (int c)

public class SlideStart extends SlideSquare (int c)

public class SlideEnd extends SlideSquare (int c)

public class HomeSquare extends Square (int c)

public class SafetyZoneSquare extends Square (int c)

public class StartSquare extends Square (int c)

public class InternalSlideSquare extends SlideSquare

public class Pawn οπου θα περιεχονται:

private int startX

private int startY

private int posX

private int posY

private int color

public void Move(int x,int y)// πρινταρει την κινηση

public void setPosX (int x)

public int getPosX ()

public int getStartX()

public void setPosY (int y)

public int getPosY ()

public getStartY()

public void setCol (int c)

public int getCol()

public void SetFacingForward (boolean b)

private boolean getFacingForward()

public String toString()

public abstract class Card οπου θα περιέχονται:

attributes καρτας:

public void SetDesc (String d)

public String getDesc ()

ποσα τετραγωνα θα κινείται η κάρτα:

public void SetMoving (int m)

public int GetMoving ()

επιστροφές τιμών:

public void kinisi( Pawn p, Board b)

public String toString ()

Ακολουθεί η κινηση καθε καρτας(για x αριθμός κάρτας)

// για καρτες 1,2,4,7,8,10,11,12

public class Cardx extends Card implements SpecialCard

public void starting(Pawn p, Board b)

public void act (Pawn p,Board b)

public Cardx () //constructor για το act

publiic interface SpecialCard

//για καρτες 3,5

public class Cardx extends Card implements SimpleCard

public void starting(Pawn p, Board b)

public void act (Pawn p,Board b)

public Cardx () //constructor για το act

publiic interface SimpleCard

public class SorryCard extends Card

public void starting(Pawn p, Board b)

public void act (Pawn p,Board b)

public SorryCard () //constructor για το act

* Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

public ArrayList<Pawn> IsChoosable() //θα ελεγχει αν μπορει να επιλεχθει ενα(η παραπανω) πιονι, θα ελεγχονται ολες οι περιπτωσεις

public void advance (Pawn p, int moveCounter) //το continue στον επομενο παιχτη.

public void Turn2 () /περιπτωση που ξαναδιαλεγει,ειτε σειρα επομενου παιχτη

public Square[][] getBlock()

public int getTurn()

public void setChoose (boolean b)

public boolean getChoose ()

public static int getposX() //οπου Χ το χρωμα του εκαστοτε παιχτη

public int turn (int p) // σειρα παιχτη

public boolean Getfold ()

public int Setfold(int f)

public boolean winner()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

τι δεν υλοποιηθηκε:

winner,fold,turn2

* Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Εδώ θα περιγράψετε τo σχέδιο υλοποίησης της προγραμματιστικής εργασίας του πακέτου view. Συγκεκριμένα, στην Α φάση του Project θα περιγράψετε το πώς σκέφτεστε να σχεδιάσετε το γραφικό περιβάλλον του πακέτου αυτού (πχ κάποιες βασικές κλάσεις, μεθόδους κλπ). Στη Β φάση, θα περιγράψετε με περισσότερες λεπτομέρειες τις κλάσεις και τις μεθόδους που χρησιμοποιήσατε.

public class SorryBoard

//θα αρχικοποιουνται οι εικόνες//

private static final int block\_h=16

private static final int block\_w=16

private Square[][] block ;

private ArrayList<Player> players=new ArrayList<Player>()

private static final int X\_PLAYER = i //αρχικοποιηση παιχτων με βαση το χρώμα

private String [] playerName;

private int Turn;

private boolean Choose

private boolean Draw

private boolean Select

private boolean Move

prvate Card pickedCard

private Pawn selectedPawn

private ArrayList<Pawn> cPawns

private Deck mainDeck

public SorryBoard ()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

τι δεν υλοποιηθηκε:

choose,draw,move,select,player name

τι αλλαξε/προστεθηκε

direction πιονιου, μεσα στην pawn με χρηση enum

μερος κωδικα απο φροντιστηριο του μαθηματος, στο JLayeredPane

* Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Σε αυτήν την ενότητα μπορείτε να συμπεριλάβετε διαγράμματα UML, και να εξηγήσετε μέσω αυτών την αλληλεπίδραση των κλάσεων (πχ μεταξύ των κλάσεων διαφορετικών πακέτων.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Η αρχικη σκεψη για την υλοποιηση της Abstract κλάσης Card ειναι η εξης:

οι καρτες χωριζονται σε 3 κατηγοριες.

Special, οπου ειναι η πλειοψηφια των καρτων που ο παιχτης εχει επιλογη στο πως θα παιξει

Simple,οπου ο παιχτης υποχρεουται να κουνησει τα πιονια προς μια κατευθυνση

Sorry!, οπου ειναι αποκλειστικα η κάρτα Sorry!

επισης οι SpecialCard και SimpleCard θα γινονται implement μεσω interfaces.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Αναφορικα με την Abstract Class Square (η Tile) η βασικη ιδεα υλοποιησης ειναι ο χωρισμος σε

Home,Start,Slide,SafetyZone και Simple Squares, για την αρχικη θεση, για την θεση εκκινησης, για τις θεσεις τσουληθρας, τις "ασφαλεις" θεσεις και τις υπολοιπες,αντιστοιχα. Επίσης, οι θεσεις "τσουληθρας" χωριζονται σε 3 επιπροσθετες κατηγοριες, οπου ειναι η αρχικη θεση της τσουληθρας, το τελος της και οι υπολοιπες ενδιαμεσες θεσεις.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Λειτουργικότητα (Β Φάση)

Στο τελικο προγραμμα αρχικοποιήται σωστα το ταμπλο, οι καρτες και οι παιχτες. τα κομματια που δεν υλοποιηθηκαν(ασχετα που γραφτηκε σχετικος κωδικας) λογω δυσλειτουργιων του controller ειναι:

κινηση πιονιων αυτοματα με βαση την καρτα

σωστη χρηση των slides

fold button (δεν χρειαζεται με βαση την υλοποιηση των χρηστων)

winner

οι παραπανω δυσλειτουργειες αντικατασταθηκαν με χειροκινητη χρηση των πιονιων,μεσω κουμπιου που βαζει ο user ποσα τετραγωνα θελει να κουνηθει το πιονι με σειρα, στα προτυπα παραδοσιακου επιτραπεζιου παιγνιου. Αν δεν υπαρχει διαθεσιμη κινηση απλα γινεται input 0.

* 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.