Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

HY252– Αντικειμενοστραφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

HY252 Project Report.

Stratego Fire vs Ice (Stratego Variation).

Δημήτριος Μακρογιάννης

CSD4676

Χειμερινό Εξάμηνο 2022-2023

Εισαγωγή.

Στο project αυτό μας ζητήθηκε να υλοποιήσουμε το μια παραλλαγή του παιχνιδιού Stratego με την αρχιτεκτονική MVC (Model View Controller).

Περιεχόμενα

OBJ

OBJ

OBJ

OBJ

OBJ

OBJ

OBJ

Εισαγωγή

Η υλοποίηση της εργασίας θα βασιστεί στο μοντέλο MVC (Model View Controller). Στόχος είναι το Controller να "κινεί τα νήματα" μεταξύ των Model και View. Παρακάτω θα αναλυθεί όλη η αρχιτεκτονική MVC και οι κλάσεις που θα χρησιμοποιηθούν στην παρούσα περίπτωση.

Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Στο πακέτο Model θα περιέχονται όλες οι κλάσεις abstract και μη που θα χρησιμοποιηθούν. Πολλές κλάσεις έχουν κάποιες υποκλάσεις οι οποίες περιγράφονται παρακάτω. Οι κύριες κλάσεις είναι οι εξής (abstract or not):

- → Board
- → Captivity
- → Player
- → Piece

Class Board and its methods

Η κλάση Board αποτελεί τον ταμπλό του παιχνιδιού δηλαδή περιέχει όλη την πληροφορία για τις θέσεις των πιονιών , πιες θέσεις είναι κενές και ποιες περιέχουν πιόνια.

Η κλάση Board περιέχει ένα enum (gameMode) το οποίο χρησιμοποιείται για τον καθορισμό των κανόνων του παιχνιδιού

Το enum (gameMode) μπορεί να πάρει τις τιμές:

→ **NORMAL** για ένα κανονικό παιχνίδι.
→ **REDUCED_ARMY** για τον κανόνα μειωμένου στρατού.
→ **TAC_0_0_11** για τον κανόνα μόνο επίθεση.
→ **BOTH** για να ισχύουν οι 2 προηγούμενοι κανόνες.

Attributes

- private Piece[ ][ ] board; (Ένας δισδιάστατος πίνακας 8x10 θέσεων ο οποίος αναπαριστά το ταμπλό του παιχνιδιού)
- private gameMode gm; (Το mode του παιχνιδιού που παίζουν οι χρήστες)

Methods

- public Board(String gm); (Constructor)
- public void setGM(String gm); (Sets enum gm field of class)
- public void setGM(gameMode gm) (Sets enum gm field of class)
- public void initBoard(gameMode gm) (Initiates Instance of class Board)
- public gameMode getGM() (Gets the gameMode of the game)
- public Piece[][] getBoard() (Gets the board of the game)
- public Piece getPiece(int x, int y); (Gets piece in the position given from the indexes)
- public void setPiece(int x, int y, Piece p); (Sets the piece given in the given coordinates on the board)
- public void deletePiece(int x, int y); (Deletes a piece from the board)

Class Captivity and its Methods

Η κλάση captvity αποτελεί το χώρο της αιχμαλωσίας κάθε παίκτη. Δηλαδή κάθε φορά που ένας παίκτης κάνει μία επιτυχή επίθεση το αντίπαλο πιόνι θα μπαίνει μέσα σε αυτό το χώρο μέχρις ότου το επαναφέρει ο κάτοχος του.

Attributes

- private ArrayList<Piece> captured; (Όλα τα πιόνια που έχουν αιχμαλωτιστεί)

- private ArrayList<Integer> sumCaptured; (Το πλήθος κάθε αιχμαλωτισμένου πιονιού)

Methods

- public Captivity(); (Constructor)

- public void initCaptivity(); (Initiates all the arraylists)

- public boolean isEmpty(); (Gets the status of the arralist)

- public void addCaptured(Piece piece); (Adds a piece to the arraylist)

- public Piece removeCaptured(Piece piece); (Removes piece of arrraylist captured)

Class Player and its Methods

Η κλάση Player αποτελεί το χώρο που αποθηκεύονται οι πληροφορίες κάθε παίχτη πόσα πιόνια εχει στη διάθεση του κάθε παίχτης, πόσες επιθέσεις έχει κάνει και πόσες από αυτές ήταν επιτυχημένες, και γενικά όλα όσα χρειάζονται για να λειτουργήσει σωστα ένα παιχνίδι.

Attributes

- private String name; (Το όνομα του κάθε παίχτη)
- private String type; (Το χρώμα του παίχτη Μπλε ή Κόκκινο)
- private int attacks, succesfullAttacks; (Επιθέσεις και επιτυχημένες επιθέσεις)
- private Captivity captivity; (Χώρος αιχμαλωσίας κάθε παίχτη)
- private int totalCaptured; (Σύνολο αιχμαλώτων)
- private ArrayList<Piece> cards (Η συλλογή πιονιών κάθε παίχτη);

Methods

- public Player(String name, String type); (Constructor)

- public void setName(String name); (Sets the field name as the String given)

- public String getName(); (Gets the field name)

- public void setType(String type); (Sets field type as the type given)

- public String getType(); (Gets the field type)

- public void setAttacks(int attacks); (Sets the field attacks as the given integer)

- public int getAttacks(); (Gets the field attacks)

- public void setSuccesfullAttacks(int succesfullAttacks); (The number of succesfull attacks of the player)

- public int getSuccesfullAttacks(); (Gets the field succesfullAttacks)

- public void addCaptured(Piece piece); (Adds the given piece to the captivity)

- public Piece removeCaptured(Piece piece); (Removes the given piece from the captivity)

- public int getTotalCaptured(); (Gets the field captivity)

- public void addCard(Piece piece); (The piece added to the collection of cards)

- public Piece getCard(int index); (Gets Piece from the collection of cards)

- public void addAttack(); (Adds one to the number of attacks)

- public void addSuccesfullAttack(); (Adds one to the number of succesfull attacks)

Class Piece and its Methods

Η κλάση Piece αναπαριστά όλα τα πιόνια που μπορούν να χρησιμοποιηθούν κατά τη διάρκεια του παιχνιδιού. Η κλάση Piece είναι abstract κλάση και έχει ως υποκλάσεις την movablePiece και την immovablePiece.

Attributes

- private int rank; (Η κατάξη του πιουνιού)
- private String name; (Το όνομα του πιονιού)
- private String type; (Το χρώμα του πιονιού)
- private ImageIcon imageUrlShow; (Μπροστά όψη της κάρτας του πιονιού)
- private ImageIcon imageUrlHide; (Πίσω όψη της κάρτας του πιονιού)
- private boolean isCaptured; (Κατάσταση αισμαλώσιας του πιονιού)
- private boolean movable; (Ικανότητα κίνησης πιονιού)

Methods

- public Piece(int rank, String name, String type, String imageUrlShow); (Constructor)

- public void setRank(int rank); (The rank of the Piece)

- public int getRank(); (Gets the field rank)

- public void setName(String name); (Sets the field name as the String given)

- public String getName(); (Gets the field name)

- public void setType(String type); (Sets field type as the type given)

- public String getType(); (Gets the field type)

- public void setImageUrlShow(String imageUrl); (Makes new ImageIcon as the image of the piece and sets the field ImageUrlShow as the ImageIcon made)

- public ImageIcon getImageUrlShow(); (Gets the field imageUrlShow)

- public void setImageUrlHide(String imageUrl); (Makes new ImageIcon as the image of the piece and sets the field ImageUrlShow as the ImageIcon made)

- public ImageIcon getImageUrlHide(); (Gets the field imageUrlHide)

- public void setIsCaptured(boolean isCaptured); (Sets field isCaptured as the boolean given)

- public boolean getIsCaptured(); (Gets the field isCaptured)

- public void setMovable(boolean movable); (Sets field movable as the boolean given)

- public boolean getMovable(); (Gets the field movable)

12

Subclass of Piece, immovablePiece and its Methods

Η υποκλάση immovablePiece κληρονομεί όλα τα attributes της υπερκλάσης και έχει επιπλέον τον δικό της constructor ,ο οποίος καλεί με τη σειρά του τον constructor της Piece. Δεν έχει μέθοδο move.

Methods

- public ImmovablePiece(String name, String type, String imageUrlShow); (Constructor)

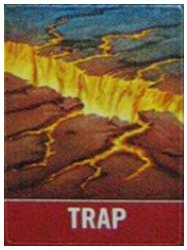
Subclass of immovablePiece, Flag and its Methods

Η κλάση **Flag** είναι immovablePiece το οποίο καθορίζει το νικητή του παιχνιδιού σε περίπτωση που αιχμαλωτιστεί.

Methods

- public Flag(String type); (Constructor)

Subclass of immovablePiece, Trap and its Methods



Η κλάση Trap είναι immovablePiece το οποίο δε μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει οποιοδήποτε από τα πιόνια του επιτεθεί εκτός απο το Dwarf.

Methods

- public Trap(String type); (Constructor)

Subclass of Piece, movablePiece and its Methods

Η υποκλάση movablePiece κληρονομεί όλα τα attributes της υπερκλάσης και έχει επιπλέον τον δικό της constructor ,ο οποίος καλεί με τη σειρά του τον constructor της Piece, και μια μέθοδο move η οποία κινεί το πιόνι σε ελεύθερη θέση του ταμπλό ή κάνει επίθεση σε πιασμένη θέση.

Methods

- public MovablePiece(int rank, String name, String type, String imageUrlShow);(Constructor)

- public void move(); (Moves the piece accross the board)

- public void capture(Piece piece); (Sends Piece to captivity and takes its place on the board)

Subclass of movablePiece, Dragon and its Methods

Η κλάση Dragon είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν.

Methods

- public Dragon(String type) (Constructor)

Subclass of movablePiece, Mage and its Methods

Η κλάση Mage είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν.

Methods

- public Mage(String type) (Constructor)

Subclass of movablePiece, Knight and its Methods

Η κλάση Knight είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν.

Methods

- public Knight(String type) (Constructor)

Subclass of movablePiece, BeastRider and its Methods

15

Η κλάση BeastRider είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν.

Methods

- public BeastRider(String type) (Constructor)

Subclass of movablePiece, Sorceress and its Methods

Η κλάση Sorceress είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτήν.

Methods

- public Sorceress(String type) (Constructor)

Subclass of movablePiece, UnleashedBeast and its Methods

Η κλάση UnleashedBeast είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτήν.

Υπάρχει μία διαφορά μεταξύ των UnleashedBeast. Αν το beast είναι στην κόκκινη ομάδα τότε γίνεται LavaBeast.

Ενώ όταν είναι στην μπλε ομάδα τοτε γίνεται Yeti.

Δεν υπάρχει κάποια διαφορα μεταξύ τους σε σχέση με τις δυνατότητες τους.

Methods

- public UnleashedBeast(String type) (Constructor)

Subclass of movablePiece, Elf and its Methods

Η κλάση Elf είναι movablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν.

Methods

- public UnleashedBeast(String type) (Constructor)

Subclass of movablePiece, specialMovablePiece and its Methods

Η υποκλάση specialMovablePiece κληρονομεί όλα τα attributes της υπερκλάσης και έχει επιπλέον τον δικό της constructor ,ο οποίος καλεί με τη σειρά του τον constructor της Piece. Κάθε κλάση που κληρονομεί αποτην SpecialImmovablePiece έχει το δικό της μοτίβο κίνησης. Άρα κάνουν override και υλοποιούν την μέθοδο move της κλάσης MovablePiece η καθεμιά με το δικό της τρόπο.

Attributes

- private boolean isSpecial;

Methods

- public SpecialMovablePiece(int rank, String name, String type, String imageUrlShow); (Constructor)

- public void setIsSpecial(boolean isSpecial); (Sets field isSpecial as the boolean given)

- public boolean GetIsSpecial(); (Gets the value of isSpecial field)

Subclass of movablePiece, Dwarf and its Methods

Η κλάση Dwarf είναι SpecialMovablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν. Επίσης είναι το μόνο Piece το οποίο δεν χάνει απο το Trap.

Methods

- public Dwarf(String type) (Constructor)

- public void move() (Overrides move method with ideal movement / attack patterns)

Subclass of movablePiece, Scout and its Methods

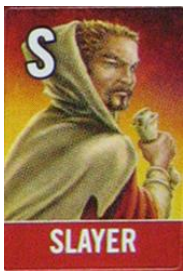
Η κλάση Scout είναι SpecialMovablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει όλα τα Pieces που ειναι κατώτερης κατάταξης από αυτόν. Επίσης είναι το μόνο Piece το οποίο μπορεί να κουνηθεί σε παραπάνω από μία ελεύθερες θέσεις στο ταμπλό του παιχνιδιού.

Methods

- public Scout(String type) (Constructor)

- public void move() (Overrides move method with ideal movement / attack patterns)

Subclass of movablePiece, Slayer and its Methods



Η κλάση Slayer είναι SpecialMovablePiece το οποίο μπορεί να κουνηθεί και μπορεί να αιχμαλωτίσει μόνο το Flag, καθώς είναι το πιόνι με τη χαμηλότερη κατάταξη. Επίσης μπορεί να σκοτώσει το Dragon μόνο όταν επιτεθεί πρώτο.

Methods

- public Slayer(String type) (Constructor)

- public void move() (Overrides move method with ideal movement / attack patterns)

Class Stratego and its Methods

Η κλάση Stratego περιέχει την main του Προγράμματος

- public static void main(String[] args); (The play button)

Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Το Controller αποτελεί τον εγκέφαλο του παιχνιδιού και τον συνδετικό κρίκο των Model και View

Attributes

- private Player p1, p2; (Οι δύο παίχτες)
- private Board b; (Το ταμπλό του παιχνιδιού)
- private int turn, round; (Η σείρα του παίχτη και ο γύρος του παιχνιδιού)
- private String gm; (Το mode του παιχνιδιού)

Methods

- public void movePiece(Piece piece, int x1, int y1, int x2, int y2); (Constructor)
- public void setP1(Player p1); (Sets the field p1 as the Player given)
- public Player getP1(); (Gets the field p1)
- public void setP2(Player p2); (Sets the field p2 as the Player given)
- public Player getP2(); (Gets the field p2)
- public Player getPlayer(Player player); (Gets the field p1 || p2)
- public void setB(Board b); (Sets the field b as the Board given)

- public Board getB(); (Gets the field b)

- public void setTurn(int turn); (Sets the field turn as the int given)

- public int getTurn(); (Gets the field turn)

- public void setRound(int round); (Sets the field round as the int given)

- public String getRound(); (Gets the field round)

- public void setGm(String gm); (Sets the field gm as the String given)

- public String getGm(); (Gets the field gm)

- public Controller(String mode); (Sets the field gm as the String given)

- public void nextTurn(); (Sets next turn)

- public void nextRound(); (Sets next round, adds one to the round counter)

- public void initGame(); (Inititates Game)

- public void endGame(); (Ends Game)

- public void restartGame(); (Restarts Game)

- public boolean swap(int x1, int y1, int x2, int y2) (if possible swaps 2 pieces)

- public void checkENd(Checks if game has ended)

- public void checkUnplayable(checks if game cannot be continued due to impossible movement from one from two players)

Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Η αρχική ιδέα του View είναι να χρησιμοποιηθεί 1 J Frame μέσα σε αυτό θα υπάρχουν 80 κουμπιά σε διάταξη 8x10 και από δίπλα 2 JPanels, το ενα θα λειτουργεί ως info box και το άλλο θα περιέχει τους ενεργούς κανόνες του παιχνιδιού είναι τελείως γενικό το πλανο αλλα σιγουρα θα υπαρχουν οι μέθοδοι redraw, move.

Στη Β φάση του Project θα υλοποιηθεί ολόκληρο το view και θα είναι πιο συγκεκριμενες οι μέθοδοι στην αναφορά,

EDIT

Έχω χρησιμοποιήσει το ταμπλό που είχα φτιάξει και στην α φάση της εργασίας. Όλη η ιδέα των γραφικών δουλεύει με μία απλη λογική 2 παράλληλων πινάκων . Έχοντας κάθε κουμπί ενα custom button handler τα επιλεγμενα κουμπια περνάνε μέσα στον πίνακα από pieces του controller και αναπρίστανται ως εικόνες πάνω στα κουμπιά των γραγικών. Όλες οι αλλαγές γίνονται μέσα στο κοντροιλλερ και απλα στρα γραφικά κάνω ένα απλό update το οποίο ξαναζωγραφίζει όλο το frame και μέσα ότι πανελ υπαρχει.

Attributes

- private JButton[][] grid;
- private boolean iconSelected;
- private JButton selected1;
- private int selected1_i, selected1_j, selected2_i, selected2_j;
- private Controller controller;
- private JPanel board_jp;
- private JPanel info_jp;
- private JPanel rules;
- private JPanel playRules;
- private JLabel top;
- private JLabel red_army;
- private JLabel full_attack;
- private JPanel statsLabJPanel;
- private JPanel statsJPanel;
- private JPanel statsJPanel1;
- private JPanel statsJPanel2;
- private JPanel statsJPanel3;
- private JLabel playerTurnJLabel;
- private JLabel statsJLabel;
- private JLabel successRateJLabel;
- private int ctr = 0;

Methods

- public View() throws IOException (Starts game makes the graphics)
- public void makeTheFrame(String gm) throws IOException (Makes the Frame)
- public void drawTheBoardPanel() throws IOException (Makes the Board panel)
- public void drawTheInfoPanel(String gm) (makes the info panel)
- private class CardListener implements MouseListener (is a custom actionlistener)
- public void setOpp() (`Sets the board for the player to play next`)
- public void update(`Updates the frame`)

Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Όλα τα uml διαγραμμάτα περιέχονται σε όλους του φακέλους.

1. Λειτουργικότητα (Β Φάση)

Υλοποιήθηκε ένα πολύ μεγάλο μέρος του πρότζεκτ το οποίο είναι η κίνηση, και ειδικότερα η σωστή κίνηση των πιονιών στο ταμπλό του παιχνιδιού με τη μόνη εξαίρεση το Scout, η οποία έχει υλοποιηθεί κατά ένα μέρος. Δεν εχει υλοποιηθει η λειτουργια της αιχμαλωσίας άρα και το revive. Επίσης στο πάνελ πληροφοριών δεν γινεται ανανέωση παρόλο που όλες οι λειτουργιες απο πιίσω ανανεωνουν τα στατιστικά.

Συμπεράσματα

Προς το παρόν είναι ένα αρκετά γενικό πλάνο της εργασίας και υπάρχουν πολλές πιθανότητες να χρησιμοποιήσω κλάσεις και μεθόδους που δεν τις έχω συμπεριλάβει και δεν τις έχω σκεφτεί καν. Οπότε το τελικό Project μπορεί να διαφέρει αισθητά από τη Φάση Α. Κάθε πακέτο/κλάση/μέθοδος έχει περιγραφή javadoc.

Edit

Η α φάση που παρέδωσα ήταν αρκετά καλή καθως είχα πάρει το άθημα και πέρσι και ήξερα σε γενικές γραμμές τι θα μου χρειαστεί άρα δεν υπαρχουν πολλες αλλαγες στις κλασεις και στις κλάσεις του μοντελου δεν υπαρχει καμία.