



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών
ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

STRATEGO PHASE A

Μιχάλης Ιεροδιακόνου

csd4773

07/12/2022

Περιεχόμενα:

Εισαγωγή

Σε αυτή την υλοποίηση το παιχνιδιού “Stratego Ice Vs Fire” θα βασιστούμε στο MVC (Model View Controller) μοντέλο. Η διεπαφή με τον χρήστη θα υλοποιείται στο View το οποίο μέσω του Controller θα συνδέεται στο Model του παιχνιδιού.

Packages

1. Model

Εδώ περιέχονται όλες οι απαραίτητες κλάσεις για την δημιουργία και ενημέρωση του παιχνιδιού.

SubPackage Coordinates:

1. Class Coordinates:

Εχω υλοποιήσει αυτή την κλάση για διευκόλυνση στη μετακίνηση πληροφοριών μεταξύ μεθόδων. Επίσης αυτή η κλάση υλοποιείται από το Πιόνια (Pieces).

Πεδία:

- `private int y;`
- `private int x;`

Μεθόδοι:

1. `public void setX(int x) {this.y = x;}` Transformer
a. Set X-coordinate
2. `public void setY(int y) {this.y = y;}` Transformer
a. Set Y-coordinate
3. `public int getX() {return x;}` Accessor
a. Get X-coordinate
4. `public int getY() {return y;}` Accessor
a. Get Y-coordinate
5. `public boolean isValid() { //TODO }` Accessor

- a. Returns true if the coordinate is in the boards (8x10) borders

SubPackage Pieces:

1. Abstract Class Piece: (extends Coordinates)

Αφαιρετική Κλάση για την υλοποίηση των πιονιών του παιχνιδιού. Αυτή η κλάση επεκτείνει την παραπάνω κλάση Συντεταγμένων άρα κάθε πιόνι έχει συντεταγμένη (x,y), την κατάταξη του, την τοποθεσία της εικόνας που του αντιστοιχεί, σε ποιά ομάδα ανήκει και αν είναι ζωντανό ή όχι.

Πεδία:

- `private int rank;`
- `private String imagePath;`
- `private boolean isBlue;`
- `private boolean isDead;`

Μεθόδοι:

1. `public void setImagePath(String imagePath) {this.imagePath = imagePath;}` Transformer
Set Image Path as sent from controller
2. `public void setDead() {isDead = true;}` Transformer
Set Piece to dead
3. `public int getRank() {return rank;}` Accessor
Returns the Rank of the piece (0-11, 0 is Flag, 11 is Trap)
4. `public String getImagePath() {return imagePath;}` Accessor
Returns the image Path so it can render it as an ImageIcon on the board
5. `public boolean isDead() {return isDead;}` Accessor
Returns true if this Piece is Dead

2. Class ImmovablePiece: (extends Piece)

Κλάση για την υλοποίηση μη κινούμενων πιονιών του παιχνιδιού. Σε αυτή την κλάση υπάγεται η Σημαία και η Παγίδα.

Πεδία:

- `private boolean isFlag;`

Μεθόδοι:

1. `public boolean isFlag() {return isFlag;} Accessor`
Returns true if this ImmovablePiece isFlag

3. Class MovablePiece: (extends Piece)

Κλάση για την υλοποίηση μετακινούμενων πιονιών του παιχνιδιού. Αυτή η κλάση επεκτείνει την παραπάνω κλάση Piece προσθέτοντας 3 νέες μεθόδους μοναδικές για τα κινούμενα πιόνια.

Πεδία:

- `private boolean wasRevived = false;`

Μεθόδοι:

1. `public void move(Coordinates newPos) throws InvalidCoordinatesException {} Transformer`
Set MovablePiece's new Coordinates from one of the possibleMoves calculated below and throw exception if they are out of bounds
2. `public void setRevived() {this.wasRevived = true;} Transformer`
Set Piece to Revived so it is not revived again
3. `public List<Coordinates> getPossibleMoves() {return new ArrayList<Coordinates>();} Accessor`
Calculates and Returns a list of coordinates (max 4) in which this piece can move to
4. `public void attack(Piece Enemy) throws DeadPieceException{} Transformer`
Simulates the attack between this MovablePiece and an Enemy Piece and sets Dead one or both depending on their respective Rank
5. `public boolean WasRevived() {return wasRevived;} Accessor`
Returns true if this Piece has already been Revived

4. Class SpecialMovablePiece: (extends Piece)

Κλάση για την υλοποίηση Ειδικών μετακινούμενων πιονιών του παιχνιδιού. Αυτή η κλάση επεκτείνει την παραπάνω κλάση Piece και η χρήση της είναι για να override των μεθόδων getPossibleMoves, attack της MovablePiece.

5. Class Slayer: (extends SpecialMovablePiece)

Κλάση για την υλοποίηση του Slayer ο οποίος υλοποιεί μια διαφορετική μέθοδο attack.

Μεθόδοι:

@Override

1. `public void attack(Piece Enemy) throws DeadPieceException{}`
Transformer
Simulates the attack between this MovablePiece and an Enemy Piece and sets Dead one or both depending on their respective Rank

6. Class Scout: (extends SpecialMovablePiece)

Κλάση για την υλοποίηση του Ανιχνευτή/Scout ο οποίος υλοποιεί μια διαφορετική μέθοδο getPossibleMoves.

Μεθόδοι:

@Override

1. `public List<Coordinates> getPossibleMoves() {return new ArrayList<Coordinates>();}` Accessor
Calculates and Returns a list of coordinates (max 4) in which this piece can move to

7. Class Dwarf: (extends SpecialMovablePiece)

Κλάση για την υλοποίηση του Dwarf ο οποίος υλοποιεί μια διαφορετική μέθοδο attack.

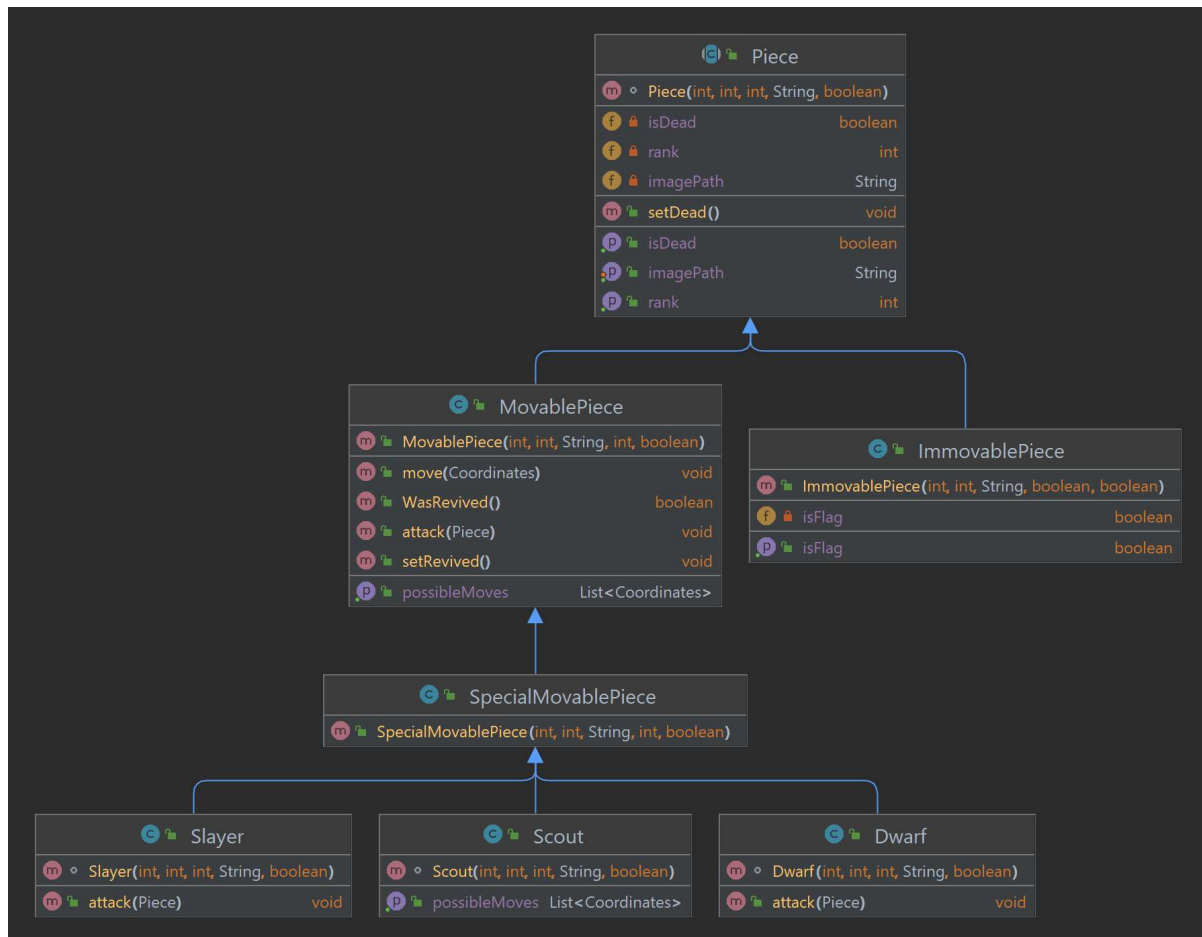
Μεθόδου:

@Override

1. `public void attack(Piece Enemy) throws DeadPieceException{}`

Transformer

Simulates the attack between this MovablePiece and an Enemy Piece and sets Dead one or both depending on their respective Rank



SubPackage Spot:

1. Class Spot: (extends Coordinates)

Κλάση για την υλοποίηση των “Θεσεων” του ταμπλό(board) του παιχνιδιού. Αυτή η κλάση επεκτείνει την παραπάνω κλάση Συντεταγμένων άρα κάθε Θέση/Spot έχει συντεταγμένη (x,y), το πιόνι σε αυτή τη θέση και αν η θέση αυτή είναι λίμνη ή όχι.

Πεδία:

- `private Piece s_Piece;`
- `private boolean isLake;`

Μεθόδοι:

1. `public void setPiece(Piece s_Piece) {this.s_Piece = s_Piece;}` Transformer
Set the Piece of this Spot to either an object Piece or null
2. `public void setLake(boolean isLake) {this.isLake = isLake;}` Transformer
Set true if Lake or false else
3. `public Piece getPiece() {return s_Piece;}` Accessor
Returns the occupying this Spot at the moment
4. `public boolean isLake() {return isLake;}` Accessor
Returns if the Spot is a Lake(Inaccessible) or not(accessible)

SubPackage Player:

1. Class Player:

Κλάση για την υλοποίηση των παικτών του παιχνιδιού. Είτε Κοκκινός () είτε Μπλε (). Κρατάνε πληροφορίες για τα πιόνια τους, ποιά είναι ζωντανά ποιά όχι, ποιά πιόνια του αντιπάλου έχουν καταλάβει

Πεδία:

- `private String m_name;`
- `private boolean isBlue;`
- `private int revival_counter;`
- `private String imagePath;`
-
- `private int m_mode;`
- `private String m_HiddenImagePath;`
- `private List<Piece> Pieces;`
- `private List<Piece> DeadPieces;`
- `private List<Piece> CapturedPieces;`

Μεθόδοι:

1. `public void initCards(int mode) {}`
Transformer
Pieces are initialized all with coordinates (0,0) to later be randomized by the board
2. `public void randomizePositions() {}`
Transformer
Pieces Coordinates are randomized depending if the Player is Blue or Red, top and bottom of the board respectively
3. `public String getM_HiddenImagePath() {return m_HiddenImagePath;}`
Accessor
Returns the path for the hidden Pieces of this Player
4. `public boolean isDefeated() {}`
Accessor
Calculates and returns if the player is defeated or not
5. `public void makeMove(Coordinates newPos) {}`
Transformer
Changes the position or state of one of its Pieces
6. `public void attacks(Piece enemyPiece) {}`
Transformer

Calls the attack method of one of its Pieces and sets defeated one or both of those pieces updating the lists

SubPackage Board:

1. Class Board:

Κλάση για την υλοποίηση του ταμπλό του παιχνιδιού αποτελούμενο από ένα δισδιάστατο πίνακα Spot[][].

Πεδία:

- `private Spot[][] spots;`

Μεθόδοι:

1. `public void initializeBoard() throws BoardNotInitialized {}`
Transformer
Initialize the Spot array with all null pointers and lakes in the appropriate positions
2. `public void placePlayer(Player p){}`
Transformer
Places the players pieces on the board
3. `public void updateBoard(Player blue, Player red){}`
Transformer
Updates the board based on both players
4. `public Spot[][] getBoard() {}`
Accessor
Returns the Spot array to update the view

SubPackage Exceptions:

Κάποια Custom Exceptions για ειδικές περιπτώσεις.

1. Class BoardNotInitializedException:

Σε περίπτωση που υπάρξει σφάλμα στην αρχικοποίηση του ταμπλό

2. DeadPieceException:

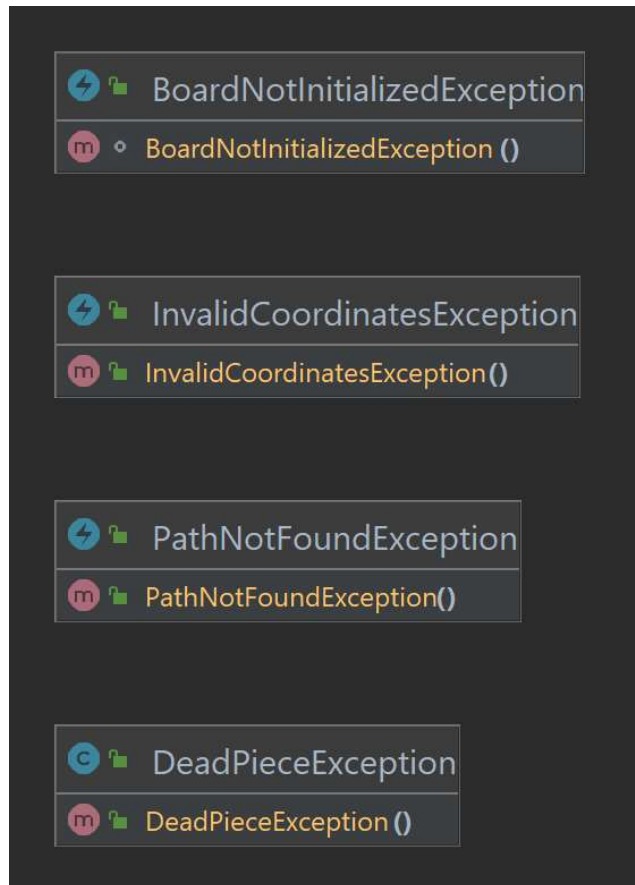
Σε περίπτωση που υπάρξει σφάλμα στην αρχικοποίηση του ταμπλό

3. InvalidCoordinatesException:

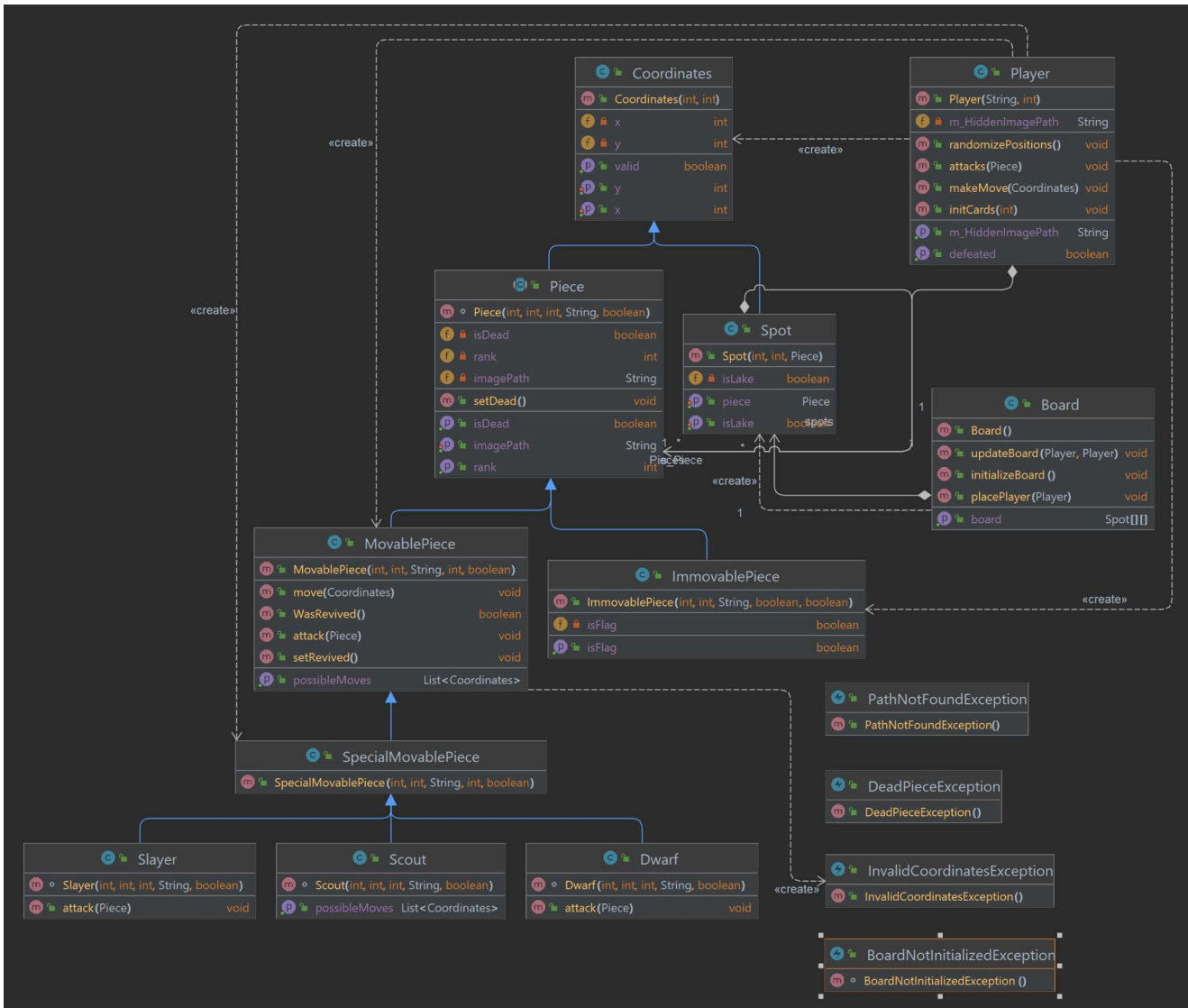
Σε περίπτωση που υπάρξει σφάλμα στην αρχικοποίηση του ταμπλό

4. PathNotFoundException:

Σε περίπτωση που υπάρξει σφάλμα στην αρχικοποίηση του ταμπλό



MODEL UML



2. Controller

1. Class Controller:

Αυτή η κλάση θα αποτελεί τον τρόπο επικοινωνίας των 2 παικτών με το ταμπλό το οποίο θα στέλνει έπειτα για τη δημιουργία των γραφικών

Πεδία:

- `private int round = 0;`
- `private boolean turnBlue = true;`
- `private Player playerBlue;`
- `private Player playerRed;`
- `private Board board;`

Μεθόδοι:

1. `public boolean isTurnBlue() {return turnBlue;}`
Accessor
Returns true if is blue players turn to play or not
2. `public int getRound() {return round;}`
Accessor
Returns the current round
3. `public Board getBoard() {return board;}`
Accessor
Returns the instance of the board
4. `public Player getPlayerBlue() {return playerBlue;}`
Accessor
Returns the instance of blue player
5. `public Player getPlayerRed() {return playerRed;}`
Accessor
Returns the instance of blue player
6. `public void nextTurn() {this.turnBlue = !this.turnBlue;}`
Transformer
Swaps between the turn of blue and red player
7. `public void nextRound() {round++;}`
Transformer
Increases round counter to go to the next round

3. View

1. Class ModSelectionWindow:

Κλάση για την δημιουργία παραθύρου διαλόγου με τον χρήστη και επιστροφή της επιλογής αυτής σαν ένα ακέραιο.

Πεδία:

- `private int m_mode;`
- `private JButton confirm;`
- `private JButton cancel;`
- `private JCheckBox reducedArmy;`
- `private JCheckBox holdW;`
- `private JPanel dialogPopUp;`

Μεθόδοι:

1. `public ModSelectionWindow() {}`
Constructor
Constructs a new Mod Selection Window which gets the mode from the user based on what is selected
2. `public int getMode() {return m_mode;}`
Accessor
Returns the calculated mode (0: no mods, 1: reduced arm, 2:only forward, 3: both)

2. Class Field: (extends JFrame)

Κλάση για την υλοποίηση της διεπαφής (αριστεράς) του πεδίου “μάχης” παίρνοντας πληροφορίες από το ταμπλό.

Πεδία:

- `private JButton[][] gridButtons;`
- `private Board f board;`

Μεθόδους:

1. `public void updateField(Board newBoard) {`
Transformer
Updates the Grid Buttons based on the newBoard reference
2. `public void drawBoard() throws PathNotFoundException {}`
Transformer
Changes this instance of Field and also adding all the components and setting the images on the buttons

3. Class Stats: (extends JFrame)

Κλάση για την υλοποίηση της διεπαφής (δεξιάς) των στατιστικών παίρνοντας πληροφορίες από τους παίκτες.

Πεδία:

- `private Player m_player;`
- `private JCheckBox l_mod1;`
- `private JCheckBox l_mod2;`
- `private boolean mod1;`
- `private boolean mod2;`
- `private int win_rate;`
- `private int revives;`
- `private int round;`
- `private int capture1;`
- `private int capture2;`
- `private int capture3;`
- `private int capture4;`
- `private int capture5;`
- `private int capture6;`
- `private int capture7;`
- `private int capture8;`
- `private int capture9;`
- `private int capture10;`
- `private int totalCaptures;`

Μεθόδους:

1. `public Stats(Player player) {}`
Constructor
Creates the Stat JFrame

2. `JPanel activeMods(int mode){}`
Transformer
Generates the Active Mods JPanel to go into the final JFrame
3. `JPanel Statistics(){}`
Transformer
Generates the Statistics JPanel to go into the final JFrame
4. `JPanel captures(Player player){}`
Transformer
Generates the Captured Units JPanel to go into the final JFrame
5. `public JFrame update(Player player){}`
Accessor
Changes this instance of Field and also adding all the components and setting the images on the buttons

4. Class View: (extends JFrame)

Η κύρια κλάση για την υλοποίηση .

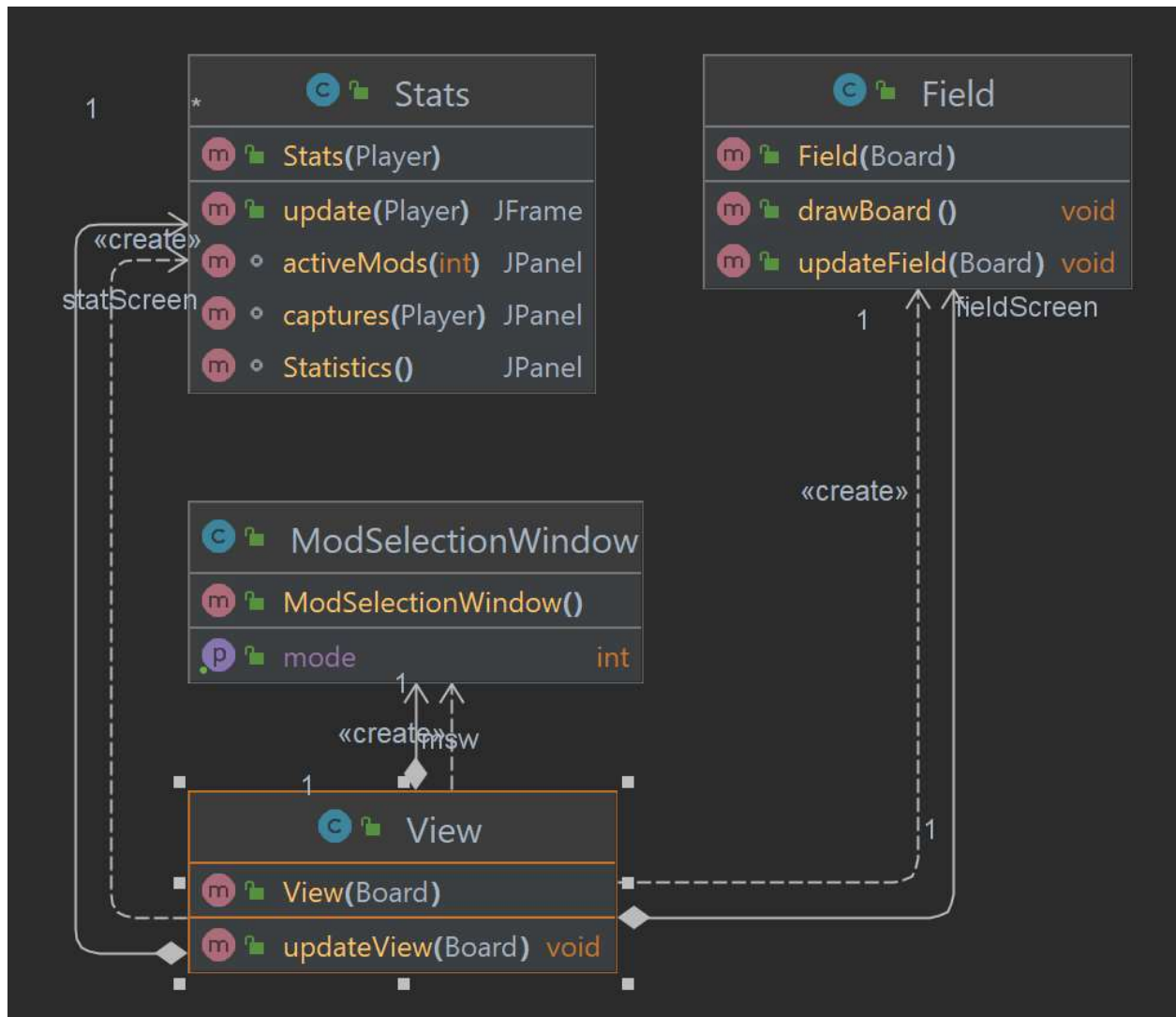
Πεδία:

- `private Stats[] statScreen;`
- `private Field fieldScreen;`
- `private Controller controller;`
- `private Player p1;`
- `private Player p2;`
- `private ArrayList<JButton> graveyard;`
- `private int mode;`
- `private ModSelectionWindow msw;`
- `private JFrame gameView;`

Μεθόδοι:

1. `public View() {}`
Constructor
Creates a new View and draws everything
2. `public void updateView(Board newBoard) {}`
Transformer
Each turn the board is updated and redrawn

VIEW UML



(Also attached as a .uml file)