Αντικειμενοστρεφής Προγραμματισμός ΗΥ-252
Project 2024-2025

Σπυρίδων Σιαχάμης - CSD5503

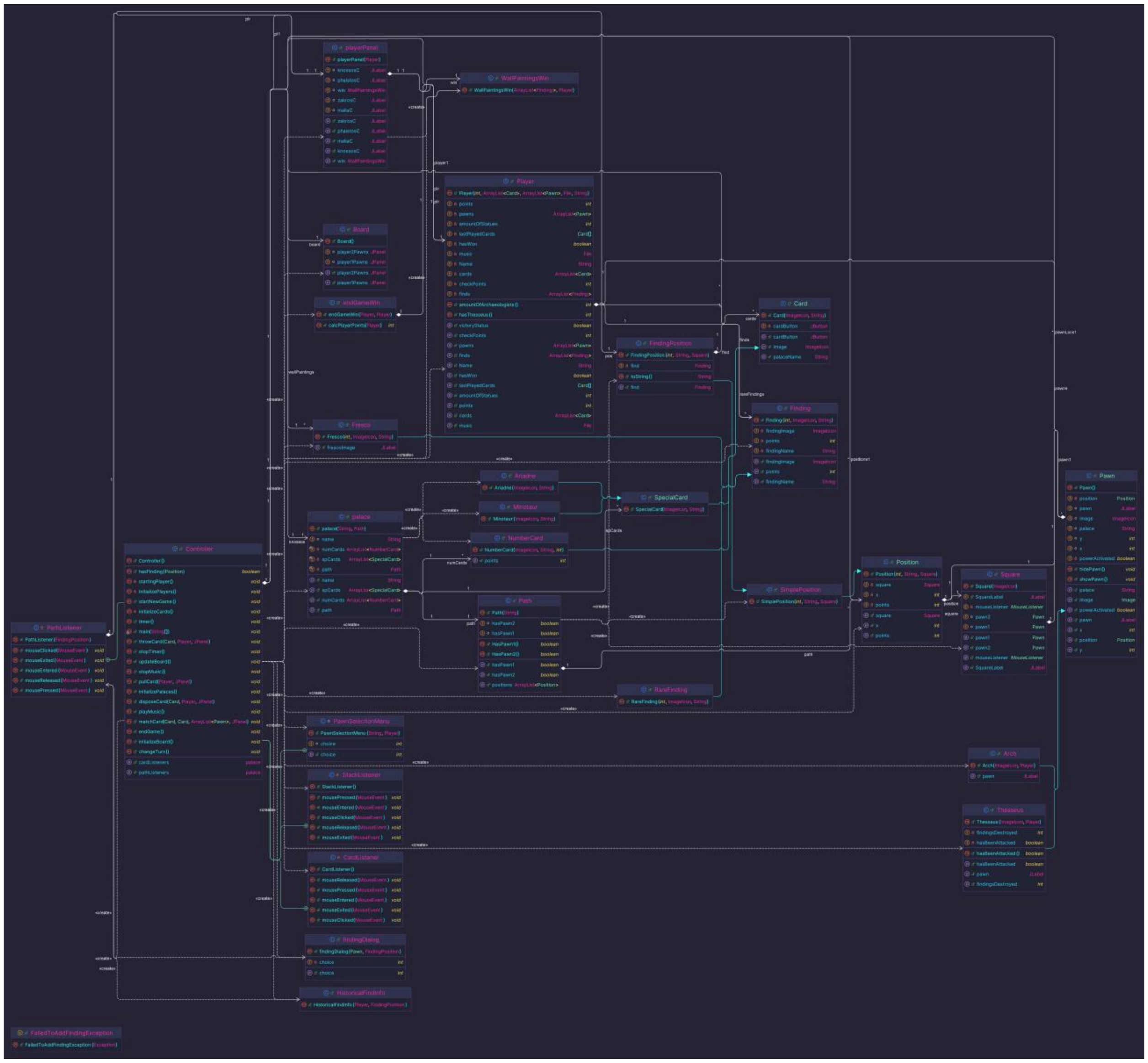
ΠΕΡΙΕΧΟΜΕΝΑ

1. Σχεδιασμός του Project

Η υλοποίηση του Project είναι βασισμένο στο μοντέλο MVC (Model View Controller), αυτό θα μας επιτρέψει να χρησιμοποιήσουμε το Controller σαν τον συνδετικό κρίκο μεταξύ του Model και του View.

Στην συνέχεια της αναφοράς θα αναλυθούν οι κλάσεις του Model και οι μέθοδοι του Controller και η εσωτερική του κλάση CardListener και τέλος οι κλάσεις του View package.

UML Diagram:



2. Model Package

Στο πακέτο του **Model**, υπάρχουν τα παρακάτω πακέτα:

2.1 Cards Package

Το πακέτο **cards**, το οποίο περιέχει τις παρακάτω κλάσεις:

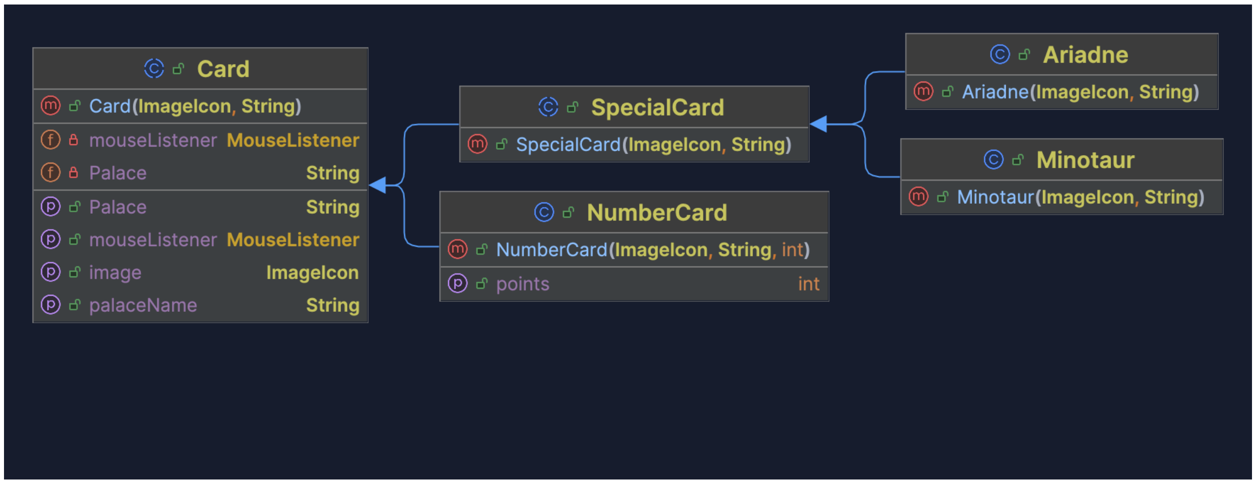
– Abstract Class Card

- Η υποκλάση της, NumberCard
- Η υποκλάση της, SpecialCard

Η **SpecialCard** έχει 2 υποκλάσεις:

- Ariadne
- Minotaur

<u>**UML Diagram**</u> των Card αντικειμένων:


Card
Card(ImageIcon, String)
mouseListener MouseListener
Palace String
Palace String
mouseListener MouseListener
image ImageIcon
palaceName String
SpecialCard
SpecialCard(ImageIcon, String)
NumberCard
NumberCard(ImageIcon, String, int)
points int
Ariadne
Ariadne(ImageIcon, String)
Minotaur
Minotaur(ImageIcon, String)

Η **Card** έχει τα εξής attributes:

- private ImageIcon minImage; // The card image is represented by a valid ImageIcon
- private String Palace; // The name of the palace that the card is associated with. Είναι ενα μη-null string.
- private MouseListener mouseListener; // A listener object for receiving mouse events on this component.

Η **Card** μας παρέχει επίσης τις εξής μεθόδους:

- **public ImageIcon getImage()** { return minImage; } // Επιστρέφει την εικόνα της κάρτας.
- public void setImage(ImageIcon minImage) // Sets the ImageIcon representing the image of the card.
- **public String getPalaceName()** // Retrieves the name of the palace associated with the card.

- **public JButton getCardButton()** // Retrieves the JButton representing the clickable button for this card.

Η **NumberCard** και η **SpecialCard** υλοποιούν την Card. Η **NumberCard** μας παρέχει την παρακάτω μέθοδο:

- **public int getPoints()** // Retrieves the points associated with the NumberCard.
- **public NumberCard(ImageIcon Image, String palace, int number)** // Initializes a NumberCard Card Class

Η **SpecialCard** δεν μας παρέχει επιπλέον μεθόδους.

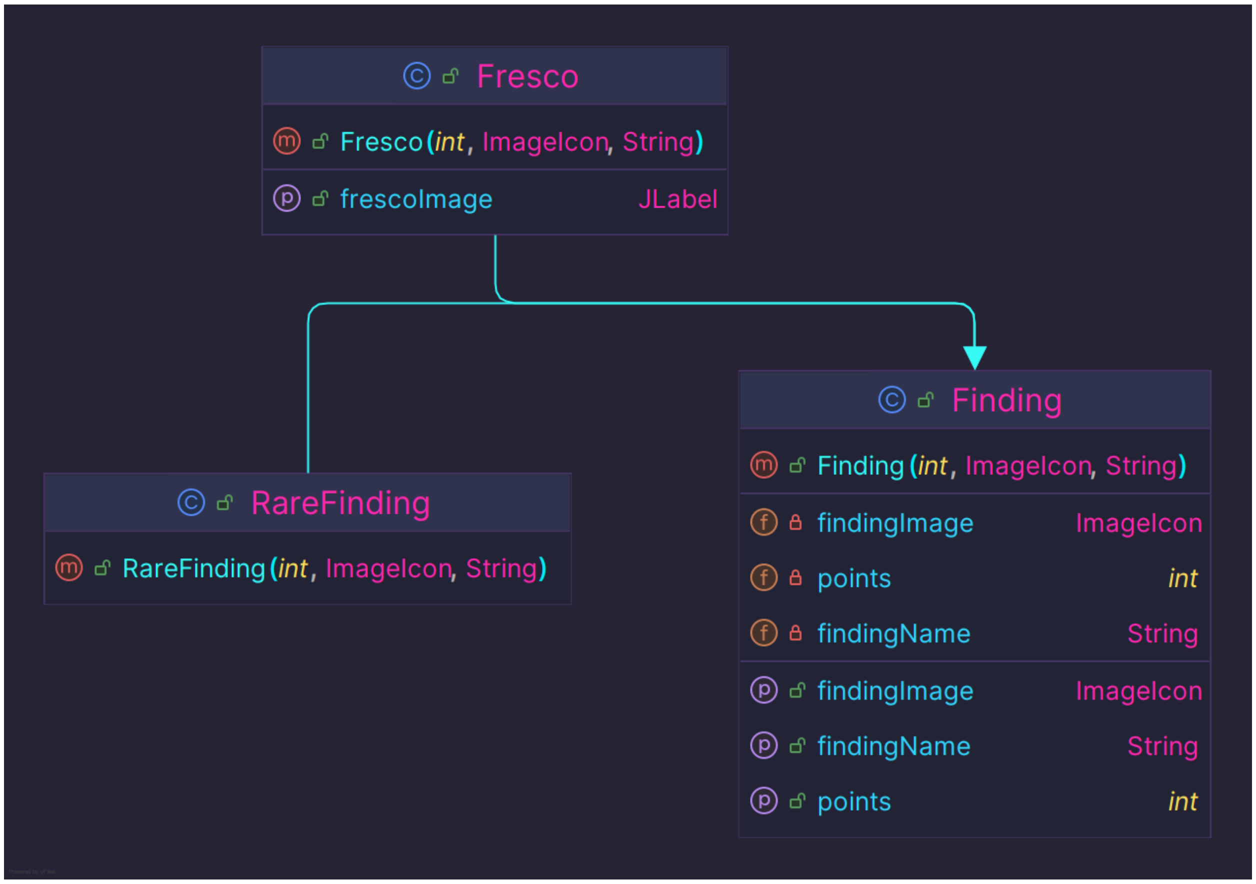
Οι κλάσεις **Minotaur** και **Ariadne** υλοποιούν την **SpecialCard** και δέν μας παρέχουν επιπλέον μεθόδους και attributes.

2.2 findings Package

Το πακέτο **findings**, το οποίο περιέχει τις παρακάτω κλάσεις:

- Finding
- Fresco
- RareFinding

<u>UML Diagram</u> του πακέτου **findings**



Η κλάση **Finding** έχει τις εξής μεθόδους και attributes:

Attributes:

- private int points; // The points of the Archaeological Finding.
- private ImageIcon findingImage; // The ImageIcon representing the image of the Archaeological Finding.
- private String findingName; // The name of the Archaeological Finding.

Μεθόδοι:

- **public int getPoints()** { return points; } // Retrieves the points of the Archaeological Finding.

- **public void setPoints(int points)** { this.points = points; } // Sets the total points of the finding.

- **public ImageIcon getFindingImage()** // Returns the Image of the Finding Object

- **public void setFindingImage(ImageIcon findingImage)** { this.findingImage = findingImage; } // Sets the image of the Archaeological Finding.

- **public String getFindingName()** // Retrieves the name of the Archaeological Finding.

Η **Fresco** και η **RareFinding** υλοποιούν την **Finding** . Η **Fresco** μας παρέχει την εξής μέθοδο:

- public JLabel getFrescoImage() // Retrieves a scaled version of the finding image as a JLabel

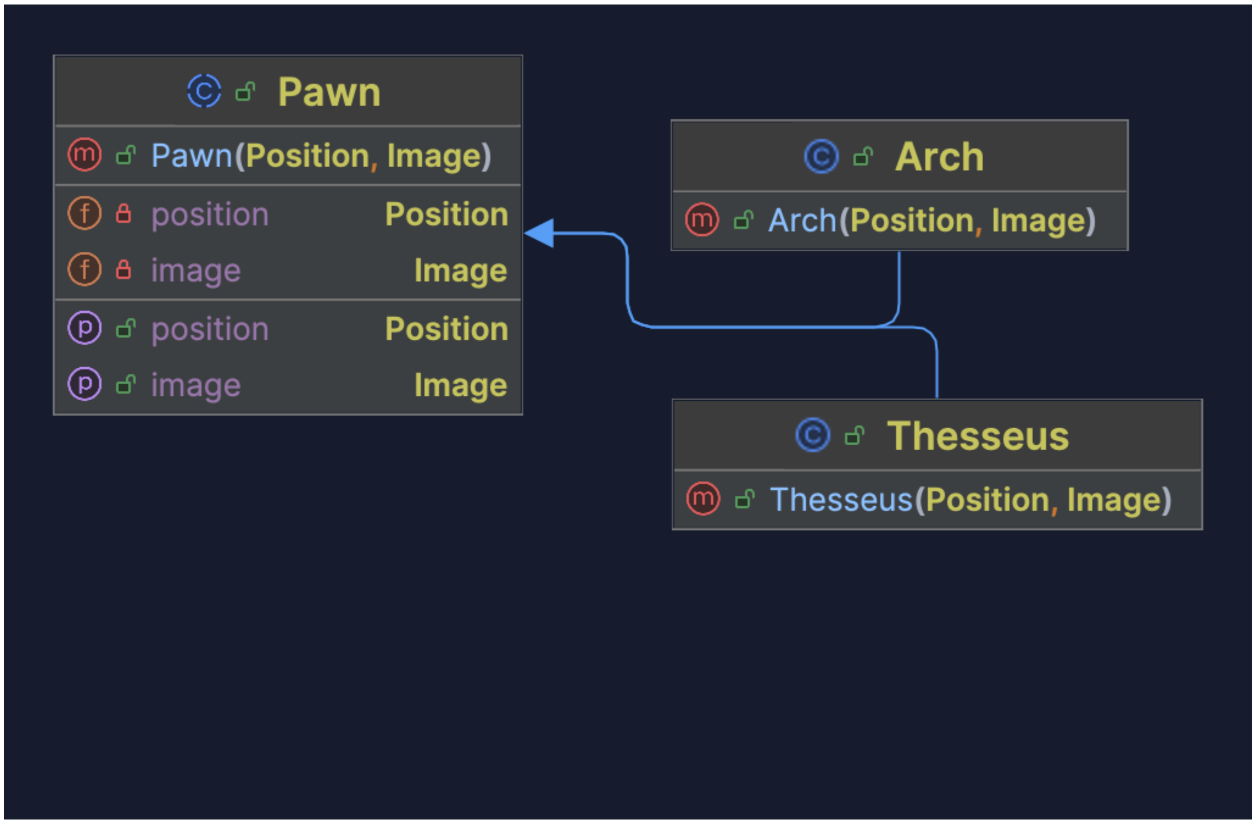
Η **RareFinding** δεν μας παρέχει παραπάνω μεθόδους.

2.3 pawns Package

Το πακέτο **pawns** μας παρέχει τις παρακάτω κλάσεις:

- Pawn
- Arch
- Thesseus

UML Diagram του πακέτου pawns:



Η κλάση **Pawn** έχει τα παρακάτω attributes:

Attributes:

- private Position position
- ImageIcon image
- JLabel pawn
- Image scaledImage
- ImageIcon question
- Image scaledQuestion
- int x,y
- ImageIcon questionIcon
- String palace
- GridBagConstraints location
- private boolean powerActivated

<u>Μέθοδοι</u>:

- **public void setPosition(Position position**) // Sets the Position object associated with this Pawn.
- **public Image getImage()** // Gets the Image object representing the visual representation of the Pawn.
- **public void setImage(Image image**) // Sets the Image object for a Pawn.
- **public JLabel getPawn()** // Retrieves the JLabel representing the Pawn's visual component.
- **public int getX()** // Retrieves the x-coordinate of the Pawn.
- **public void setX(int x)** // Sets the x-coordinate of the Pawn.
- **public int getY()** // Retrieves the y-coordinate of the Pawn.
- **public void setY(int y)** // Sets the y-coordinate of the Pawn
- **public String getPalace()** // Retrieves the palace name associated with the Pawn.
- **public void setPalace(String palace)** // Sets the palace name associated with the Pawn.
- **public void hidePawn()** // Hides the Pawn by setting its icon to a question mark image.
- **public void showPawn()** // Reveals the Pawn by setting its icon to the scaled image associated with it.
- **public boolean isPowerActivated()** // Retrieves whether the Pawn's power has been used once.
- **public void setPowerActivated(boolean powerActivated)** // Sets the activation state of the Pawn's power.

Η υποκλάση **Arch** μας παρέχει την παραπάνω μέθοδο:

- **public JLabel getPawn()** // Retrieves the JLabel representing the Arch pawn.

Η υπόκλαση **Thesseus** μας παρέχει τις παρακάτω μεθόδους και Attributes:

Attributes:

- private int findingsDestroyed = 0;
- private boolean hasBeenAttacked = false;

Μεθόδοι:

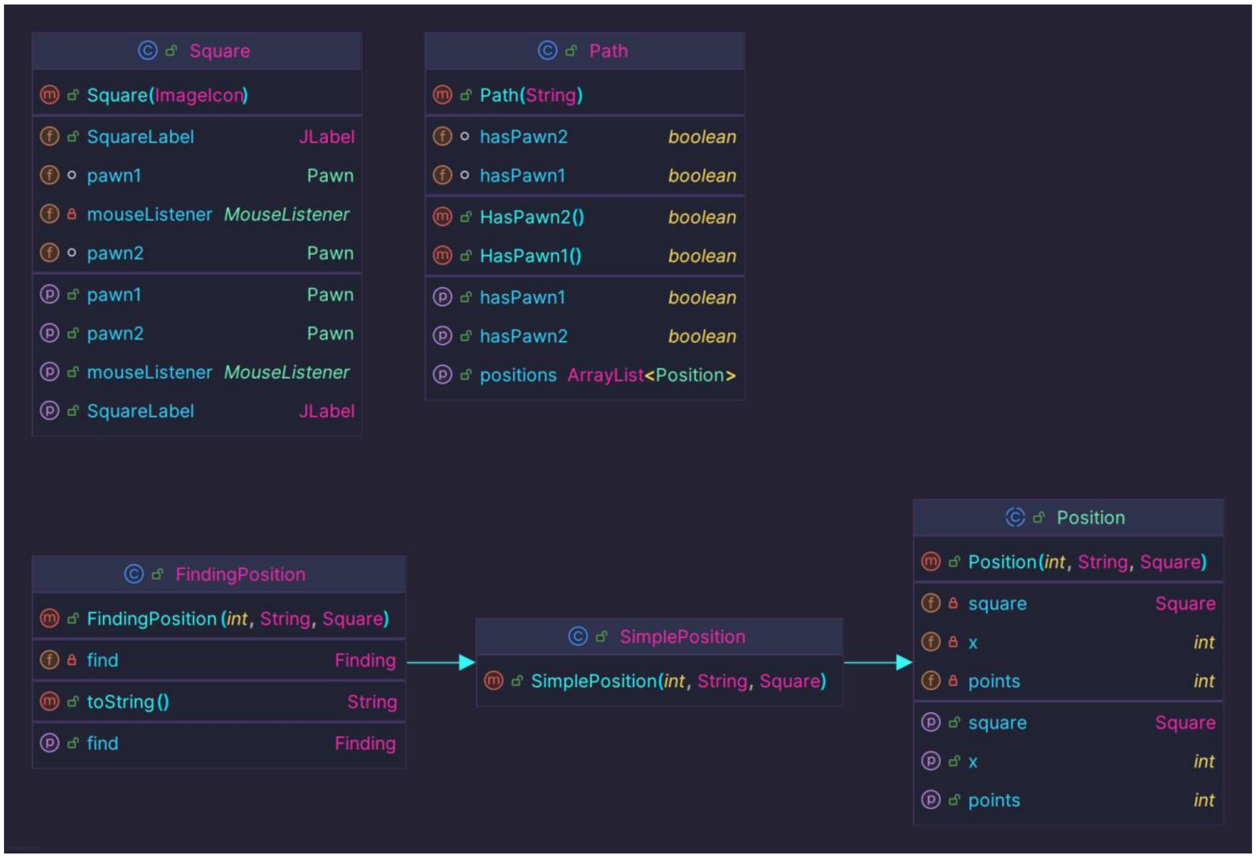
- **public JLabel getPawn()** // Retrieves the JLabel representing the visual component of the Thesseus pawn.
- **public int getFindingsDestroyed()** // Retrieves the number of findings destroyed by this Thesseus pawn.
- **public void setFindingsDestroyed(int findingsDestroyed)** // Sets the number of findings destroyed by this Thesseus pawn.
- **public boolean hasBeenAttacked()** // Retrieves whether the Thesseus pawn has been attacked by a {@link Minotaur} card of the enemy player.
- **public void setHasBeenAttacked(boolean hasBeenAttacked)** // Sets the attacked state of the Thesseus pawn.

2.4 positions Package

Το **positions** package μας παρέχει τις παρακάτω κλάσεις:

- Path
- Position
- SimplePosition
- FindingPosition
- Square

<u>UML Diagram</u> του **positions** package:



Η κλάση **Path** μας παρέχει τις παρακάτω μεθόδους και attributes:

Attributes:

- private ArrayList<Position> path // Positions of path
- private String pathName // Name of path
- private ArrayList<Pawn> pathPawns // Pawns of path
- boolean hasPawn1 = false; // Boolean variable to designate if path has a pawn of player1
- boolean hasPawn2 = false; // Boolean variable to designate if path has a pawn of player2

Μεθόδοι:

- **public ArrayList<Position> getPositions()** // Returns the ArrayList of Position objects comprising the Path.

- **public boolean HasPawn1()** // Checks if a pawn of player1 is present on this path.

- **public void setHasPawn1(boolean hasPawn1)** // Sets the Pawn of player1 on this path.

- **public boolean HasPawn2()** // Checks if a pawn of player2 is present on this path.

- **public void setHasPawn2(boolean hasPawn2)** // Sets the Pawn of player2 on this path.

Η κλάση **Position** έχει τα εξής attributes και mεθόδους:

Attributes:

- private int x; // x – coordinate of position
- private String path; // Path name that the position belongs in
- private int points; // Points of position
- private Square square; // The Square object associated with this Position.

Μεθόδοι:

- **public int getX()** // Retrieves the x-coordinate of the Position object.

- **public void setPoints(int points)** // Sets the points for the current Position.

- **public int getPoints()** // Retrieves the points associated with this Position object.

- **public Square getSquare(**) // Retrieves the Square associated with this Position.

Η κλάση **SimplePosition** και **FindingPosition** υλοποιούν την **Position**.

Η **SimplePosition** δεν παρέχει παραπάνω μεθόδους ή attributes.

Η **FindingPosition** εχει τα εξής:

Attributes:

- Private Finding find; // Finding object associated with the FindingPosition

Μεθόδοι:

- **public FindingPosition(int x, String path, Square square)** { super(x, path, square); } // Initializes a FindingPosition object with the given x-coordinate, Path object, points value, Square object, and Finding object.

- **public Finding getFind()** // Retrieves the Finding object associated with the FindingPosition instance.

- **public void setFind(Finding find)** // Sets the Finding object associated with the FindingPosition instance.

Η κλαση **Square** έχει τα εξής:

Attributes:

- ImageIcon SquareImage; // The image of the Square object
- public JLabel SquareLabel
- private MouseListener mouseListener
- Pawn pawn1;
- Pawn pawn2;

Μεθόδοι:

- **public Square(ImageIcon squareImage)** {SquareImage = squareImage;} // Initializes the Square with a given ImageIcon.

- **public JLabel getSquareLabel()** // Retrieves the JLabel representing the square.

- **public void setMouseListener(MouseListener mouseListener)** // Sets the MouseListener for this square.

- **public Pawn getPawn1()** // Retrieves the Pawn of player1 on this square.

- **public void setPawn1(Pawn pawn)** // Sets the Pawn of player1 on this square.

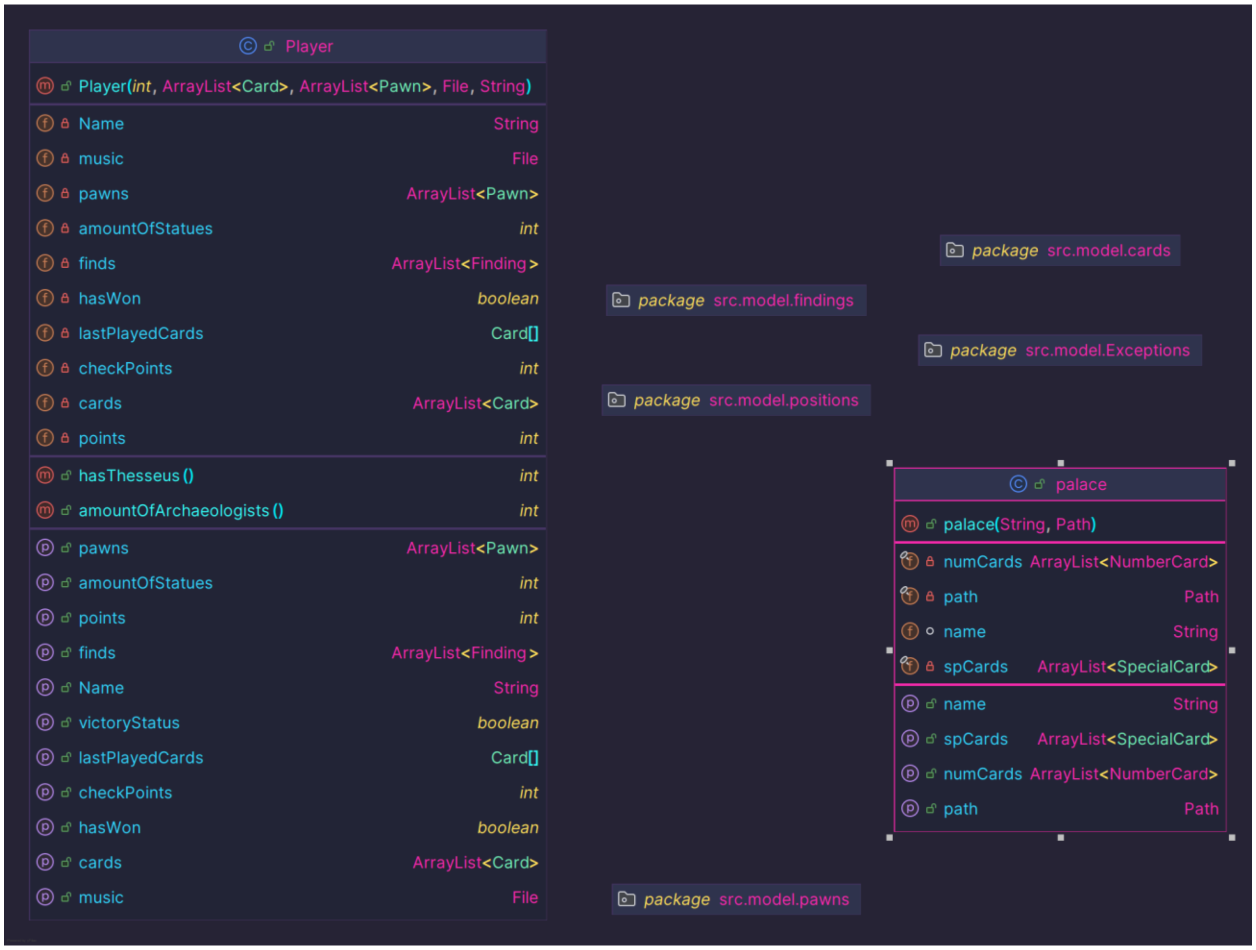
- **public Pawn getPawn2()** // Retrieves the Pawn of player2 on this square.

- **public void setPawn2(Pawn pawn2)** // Sets the Pawn of player2 on this square.

2.5 Palace Class

Η κλάση **Palace** δεν ανήκει σε εσωτερικό package.

<u>UML Diagram</u> της Palace:



Η κλάση **palace** περιέχει τα εξής Attributes:

- private int totalCards ; // Total Card amount of the palace
- String name; // Name of the palace
- private final Path path; // Path Array of positions
- private final ArrayList<NumberCard> numCards; // ArrayList of NumberCard objects of the Palace
- private final ArrayList<SpecialCard> spCards; // ArrayList of SpecialCard objects of the Palace

Περιέχει επιπλέον τις παρακάτω **μεθόδους**:

- **public palace(String name, Path path)** // Constructs a new Palace object with a specified name and associated path.

- **public ArrayList<NumberCard> getNumCards()** { return numCards;} // Retrieves the list of special cards associated with the palace.

- **public ArrayList<SpecialCard> getSpCards()** { return spCards; }public Path getPath() { return path;} // Returns the Path object associated with this palace.

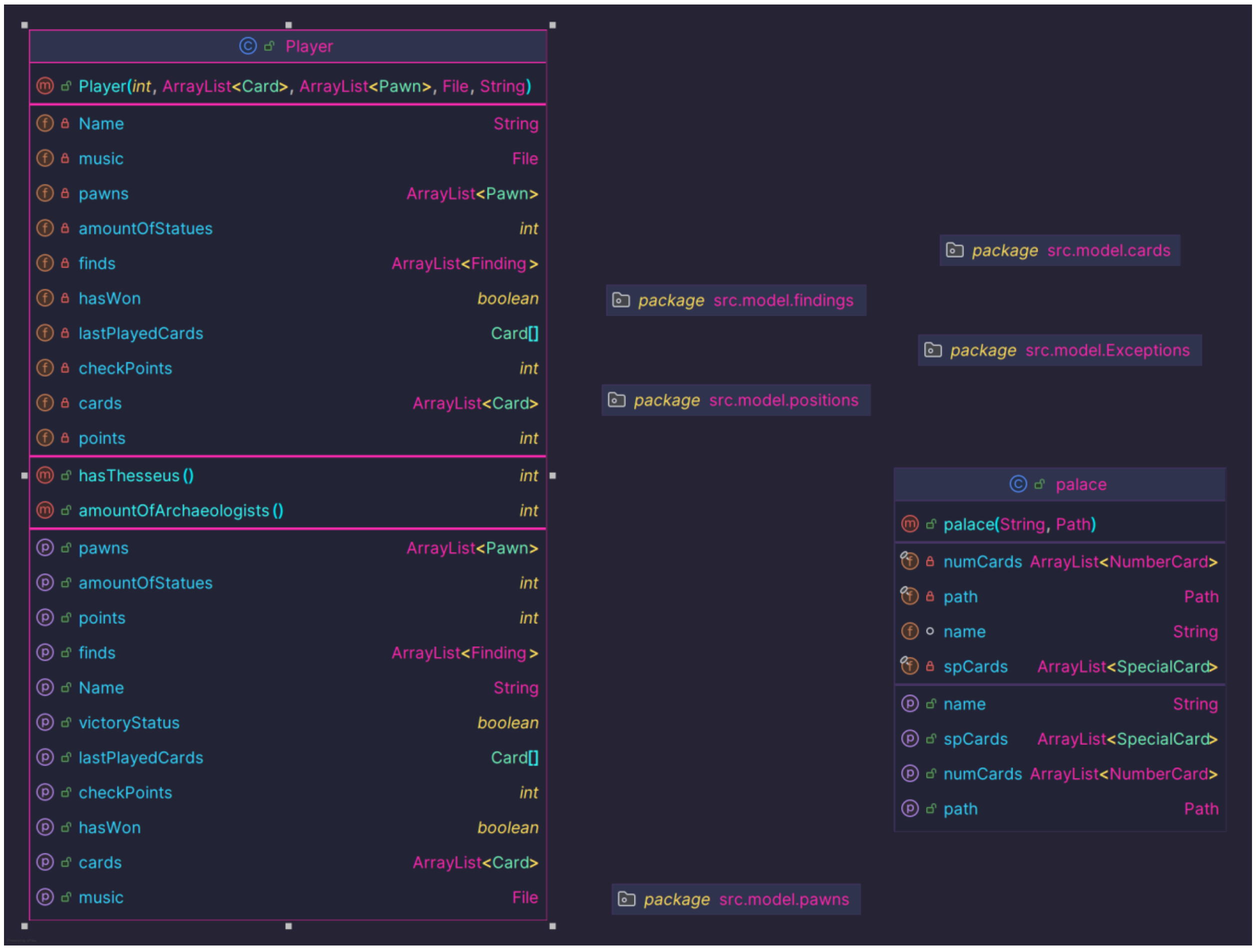
- **public String getName()**{return name; } // Retrieves the name associated with the palace.

- **public Path getPath()** // Returns the Path object associated with this palace.

2.6 Player Class

Η **Player** κλάση δεν ανοικεί σε κανένα sub-package του **Model** package.

UML Diagram του Player Class



```
Player
Player(int, ArrayList<Card>, ArrayList<Pawn>, File, String)
Name                    String
music                   File
pawns                   ArrayList<Pawn>
amountOfStatues         int
finds                   ArrayList<Finding>
hasWon                  boolean
lastPlayedCards         Card[]
checkPoints             int
cards                   ArrayList<Card>
points                  int
hasThesseus()           int
amountOfArchaeologists() int
pawns                   ArrayList<Pawn>
amountOfStatues         int
points                  int
finds                   ArrayList<Finding>
Name                    String
victoryStatus           boolean
lastPlayedCards         Card[]
checkPoints             int
hasWon                  boolean
cards                   ArrayList<Card>
music                   File
```

```
package src.model.cards
package src.model.findings
package src.model.Exceptions
package src.model.positions
package src.model.pawns
```

```
palace
palace(String, Path)
numCards   ArrayList<NumberCard>
path       Path
name       String
spCards    ArrayList<SpecialCard>
name       String
spCards    ArrayList<SpecialCard>
numCards   ArrayList<NumberCard>
path       Path
```

Η **Player** περιέχει τα παρακάτω Attributes:

Attributes:

- private String Name; // Represents the name of a Player.

- private int points; // Represents the total points accumulated by a Player.

- private ArrayList<Finding> finds; // Holds a collection of archaeological findings associated with the player.

- private ArrayList<Card> cards; // Represents the collection of cards held by the player.

- private ArrayList<Pawn> pawns; // A list of Pawn objects associated with a player.

- private boolean hasWon; // Indicates whether the player has won the game.

- private File music; // Represents the music file associated with the player.
- private int amountOfStatues // Amount of snake statues of the player.
- private int checkPoints = 0; // Amount of checkpoints the player has.
- private Card[] lastPlayedCards // Last played card of each palace.

Οι **μεθόδοι** που μας παρέχει η Player κλάση είναι οι εξής:

- **public Player(int points, ArrayList<Card> cards, ArrayList<Pawn> pawns, File music, String Name)** // Initializes a Player Object.

- **public int getPoints()** // Retrieves the total points of the Player.

- **public void setPoints(int points)** // Sets the points of the Player.

- **public ArrayList<Finding> getFinds()**} // Retrieves the list of archaeological findings associated with the player.

- **public ArrayList<Card> getCards()** // Retrieves the list of cards associated with the player.

- **public void setCards(ArrayList<Card> cards)** // Sets the cards of the Player.

- **public ArrayList<Pawn> getPawns()** // Retrieves the list of Pawns associated with the player.

- **public int amountOfArchaiologists()** // Calculates the number of archaeological pawns in the player's collection.
- **public int hasThesseus()** // Checks if any of the pawns in the player's list are instances of Thesseus.

- **public File getMusic()** // Retrieves the music file associated with the player.

- **public String getName()** // Gets the name of the player.

- **public void setName(String name)** // Sets the name of the player

- **public void setCheckPoints(int checkPoints)** // Sets the number of check points for the player.

- **public int getCheckPoints()** // Retrieves the current number of check points accumulated by the player.

- **public Card[] getLastPlayedCards()** // Retrieves the array of the last played cards by the player.

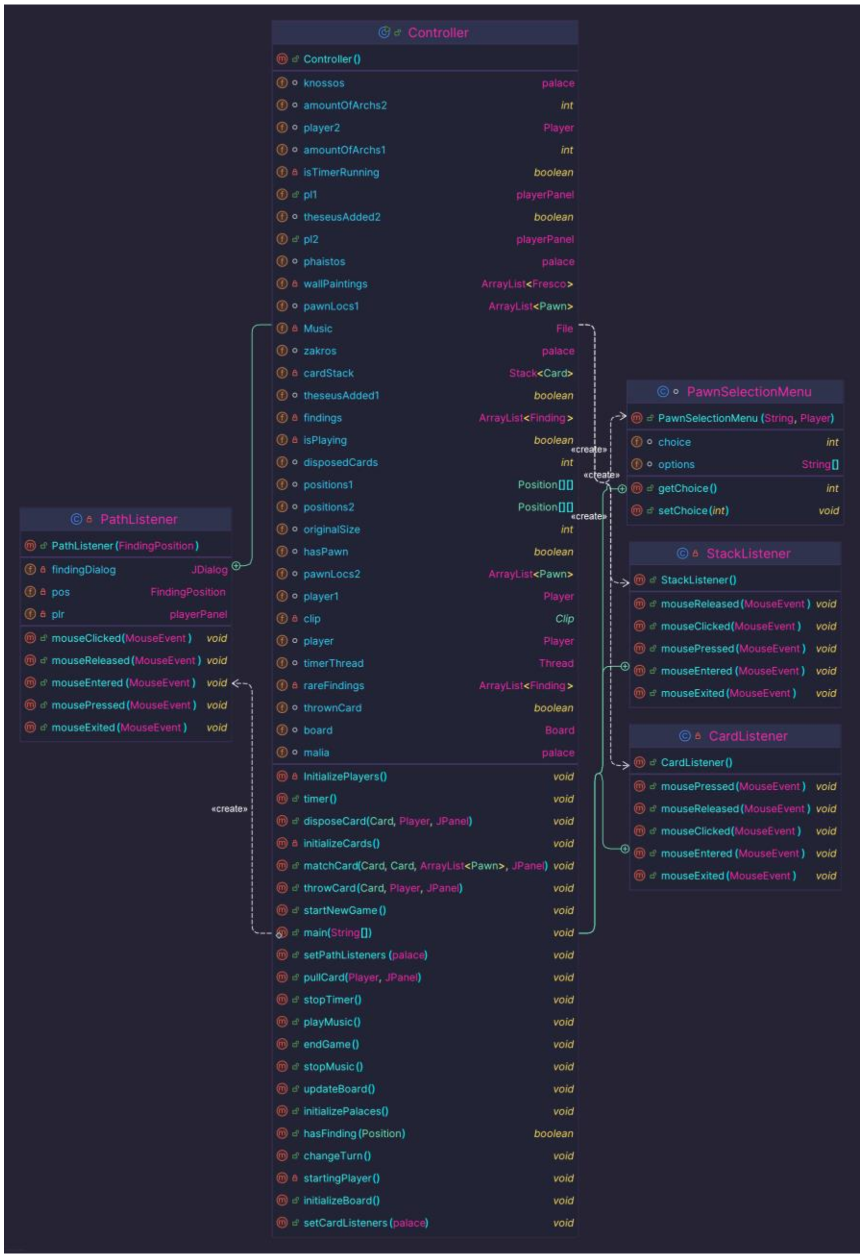
- **public int getAmountOfStatues()** // Retrieves the number of statues collected by the player.

- **public void setAmountOfStatues(int amountOfStatues)** // Sets the number of statues collected by the player.

3. Controller Class

Η **Controller** κλάση είναι ο συνδετικός κρίκος που θα ενώνει το View και το Model και θα επιτρέπει στην επικοινωνία μεταξύ τους. Επίσης θα ελέγχει την λογική του παιχνιδιού και θα υλοποιεί διάφορες βασικές λειτουργίες του.

Το **<u>UML Diagram</u>** της Controller κλάσης είναι εξής:



Το **Controller** μας παρέχει τα παρακάτω Attributes:

Attributes:

- Player player1; // Player 1
- Player player2;// Player 2
- palace knossos; // palace object with the name knossos.
- palace malia; // palace object with the name malia.
- palace zakros; // palace object with the name zakros.

- palace phaistos; // palace object with the name phaistos.
- private Stack<Card> cardStack; // Stack of cards that will contain the cards of the
- palaces not in the player’s arraylist
- private ArrayList<Finding> rareFindings(); //RareFindings ArrayList of the game that contains all the Findings
- private ArrayList<Finding> findings
- private ArrayList<Fresco> wallPaintings
- public playerPanel pl1
- public playerPanel pl2
- private File Music; // Music that will play during the runtime of the game
- private Clip clip; // Clip object for the music
- Board board
- Player player
- int disposedCards
- boolean thrownCard
- Position[][] positions1
- Position[][] positions2
- int originalSize;
- private boolean isTimerRunning;

Το **Controller** επίσης μας παρέχει επίσης τις παρακάτω μεθόδους:

- public Controller() // Constructs a src.Controller instance and initializes the game state.
- public void initializePalaces() // Initializes the palaces with their respective names and paths.
- private void InitializePlayers() // Initializes the players for the game.
- private void startingPlayer() // Randomly selects the starting player and sets their music file.
- public void playMusic() // Plays the music associated with the current player in a continuous loop.
- public void stopMusic() // Stops the currently playing music
- public void changeTurn() // Switches the turn to the next player and updates the music accordingly.
- public void updateBoard() // Updates the game board to reflect the current state of the game, including information about the available cards, the players' checkpoints, points, statues, and whose turn it is.

- private void initializeCards() // Initializes the card stack from the cards of the palaces and assigns cards to the players.
- public void initializeBoard() // Initializes the game board by setting up all the necessary prerequisites for the game to begin.

INNER CLASS: class PawnSelectionMenu // A dialog window that allows the player to select a pawn for a specified path.

- public PawnSelectionMenu(String path, Player plr) // Constructs a PawnSelectionMenu that allows the player to select a pawn.
- public void startNewGame() // Starts a new game by initializing the game board, initializing the players, and selecting the starting player.
- public void matchCard(Card c1, Card c2, ArrayList<Pawn> pawns, JPanel panel) // Determines whether a pawn can move based on the current and previous cards played
- public void throwCard(Card c1, Player player, JPanel panel) // Throws a card for the player and places a pawn on the corresponding path of the palace or moves the pawn if a pawn has already been added
- public void pullCard(Player player, JPanel panel) // Pulls a new card from the card stack and adds it to the player's hand.
- public void disposeCard(Card c1, Player player, JPanel panel) // Disposes of a card from the player's hand and removes it from the player panel.

INNER CLASS: private class CardListener // CardListener is a private inner class that implements MouseListener to handle mouse events

INNER CLASS: private class StackListener // A mouse listener for handling click events on the card stuck button in the game.

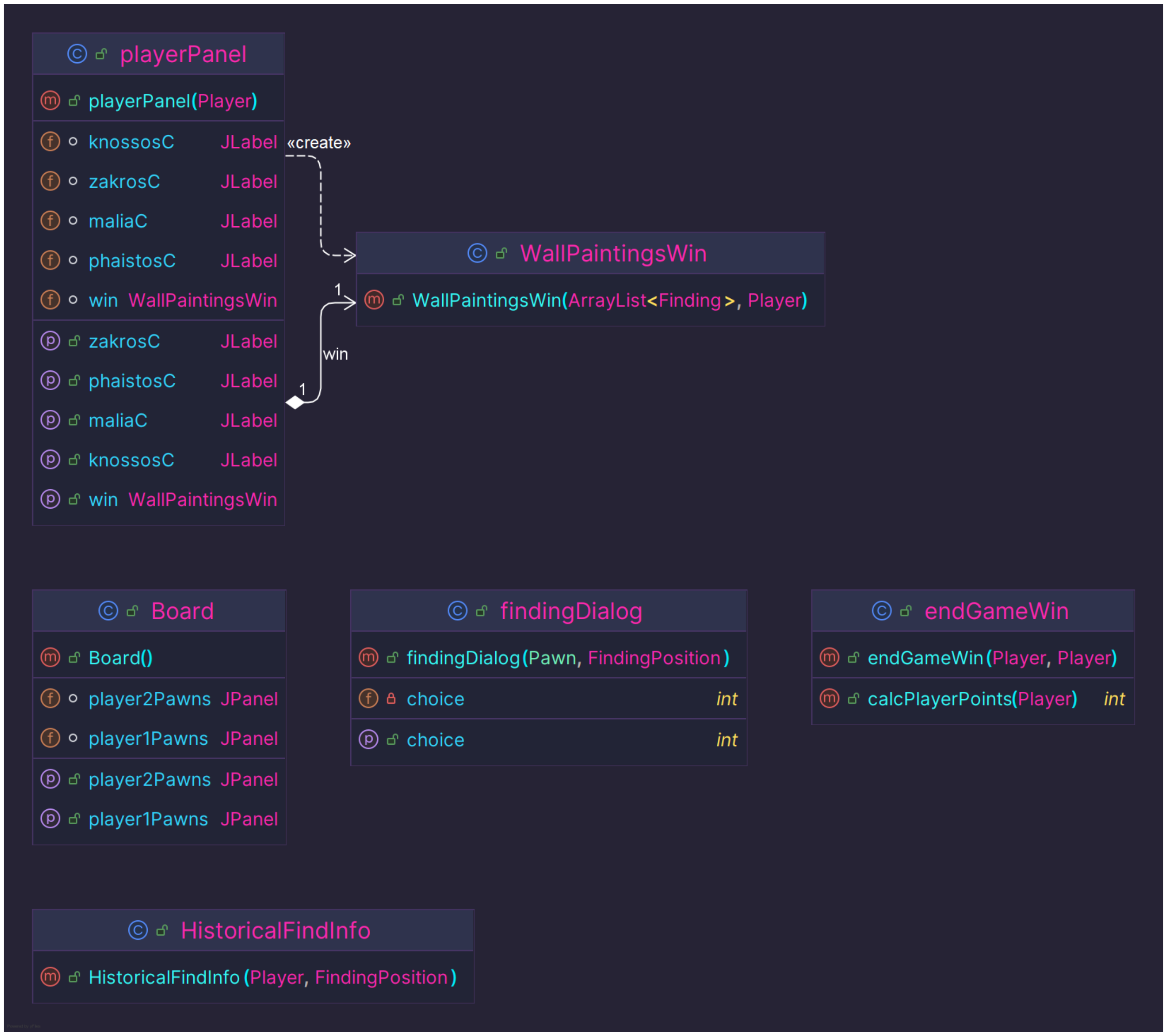
INNER CLASS: private class PathListener // A mouse listener for handling interactions with positions in the game.

- public void endGame() // Ends the current game session and displays the end game window with the results.
- public void setCardListeners(palace Palace) // Attaches mouse listeners to each card in the specified palace.
- public void setPathListeners(palace Palace) // Sets up mouse listeners for specific positions along the path of the given palace.
- public static void main(String[] args) // The main method serves as the entry point for the application, initializing and starting a new game session.

- public void timer() // Starts a countdown timer for the current player's turn and updates the game UI accordingly.
- public void stopTimer() // Stops the currently running timer thread if it is active, interrupting its execution.

4. View Package

Το **View** package είναι υπεύθυνο για τα γραφικά του παιχνιδιού μας και συνεργάζεται μαζί με τον Controller για να επικοινωνήσει μαζί με το Model. **Το UML Diagram του View:**



Το **View** μας παρέχει τις παρακάτω κλάσεις:

- Board
- endGameWin
- findingDialog
- HistoricalFindInfo
- playerPanel
- WallPaintinsWin

Το **board** μας παρέχει τα παρακάτω attributes:

- JLabel timer
- public JMenuBar menuBar
- public JMenuItem newGame
- public JMenuItem Save
- public JMenuItem Continue
- public JMenuItem Exit
- public JMenuItem mute
- public JLayeredPane Dashboard
- public JButton cardStackBut
- public JLabel stackInfo
- public JLayeredPane layeredPathPane
- public JPanel paths
- JPanel player1Pawns
- JPanel player2Pawns
- public GridBagConstraints gbc

Επίσης μας παρέχει τις παρακάτω μεθόδους:

- public JPanel getPlayer2Pawns() // Returns the panel containing the pawns for player 2.
- public JPanel getPlayer1Pawns() // Returns the panel containing the pawns for player 1.

Το **endGameWin** μας παρέχει την παρακάτω μεθόδο και έχει ως ρόλο να δώσει πληροφορίες στο τελος του παιχνιδιού στους παίκτες για τον νικητή:

- **public int calcPlayerPoints(Player plr)** // Calculates the total points for a given player based on their current points, the number of statues collected, and other game-specific criteria.

Το **findingDialog** μας παρέχει το παρακάτω attribute και εχει ως ρόλο τον έλεγχο τον ανασκάφων η της καταστροφής των ευρημάτων:

- private int choice // The choice of action for the finding.

Επιπλέον μας παρέχει με την εξής μέθοδο:

- public int getChoice() // Returns the player's choice from the dialog.

Το **HistoricalFindInfo** έχει ως ρόλο την παροχή των πληροφοριών των ευρημάτων στον παίκτη χρησιμοποιώντας πληροφορίες απο ένα csv_greek.csv αρχείο που περιέχει τις πληροφορίες αυτες

Μας παρέχει με την παρακάτω μεθόδο:

- **public HistoricalFindInfo(Player plr, FindingPosition pos)** // provides a dialog window displaying historical information

Το **playerPanel** έχει ως ρόλο να παίζει το κύριο κομμάτι του γραφικού περιβάλλοντος του παιχνιδιού μας, καθώς παρέχει στον παίκτη πληροφορίες για τις κάρτες που έχει, την τελευταία κάρτα που έπαιξε σε κάθε ανάκτορο και παρέχει εύκολη πρόσβαση στις τοιχογραφίες που έχει ήδη βρει.

Μας παρέχει με τα παρακάτω attributes:

- Player plr
- JPanel panel
- public JLabel playerStats
- public JPanel cardsPanel
- JPanel score
- public JPanel lastPlayedCardsPanel
- public JLabel timerInstance
- WallPaintingsWin win
- public JLabel diskos
- public JLabel ring
- public JLabel kosmima
- public JLabel ruto
- JLabel knossosC
- JLabel maliaC
- JLabel phaistosC

- Label zakrosC

Μας παρέχει επίσης με τις παρακάτω μεθόδους:

- public playerPanel(Player player) // Constructs a new player panel for a given player. Initializes the panel layout, sets up player statistics, cards, and interactive components like buttons.
- public void setWin(WallPaintingsWin win) // Sets the WallPaintingsWin object associated with the player panel.
- public JLabel getZakrosC() // Gets the Zakros palace label.
- public JLabel getPhaistosC() // Gets the Phaistos palace label.
- public JLabel getMaliaC() // Gets the Malia palace label.
- public JLabel getKnossosC() // Gets the Knossos palace label.

Τέλος, το **WallPaintingsWin** μας παρέχει με το παρακάτω attribute:

- private JPanel paintings // JPanel which contains all the Fresco paintings the player discovers.

Έχει ως ρόλο την παροχή εύκολης πρόσβασης στις τοιχογραφίες που φωτογραφίζει ο παίκτης κατά την διάρκεια του παιχνιδιού ώστε να μπορεί να παρακολουθήσει τις τοιχογραφίες αυτες.

Δεν μας παρέχει με παραπάνω μεθόδους.

ΤΕΛΟΣ ΑΝΑΦΟΡΆΣ