CS120 - INTRODUCTION TO
OBJECT ORIENTED PROGRAMMING

QUARTO

by Hrachya Hovakimyan
and Khachik Zakaryan

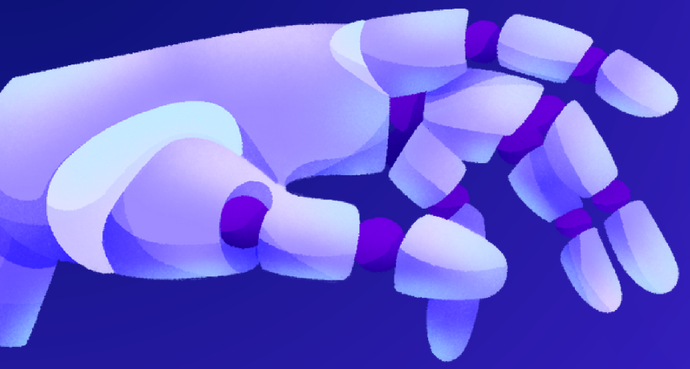
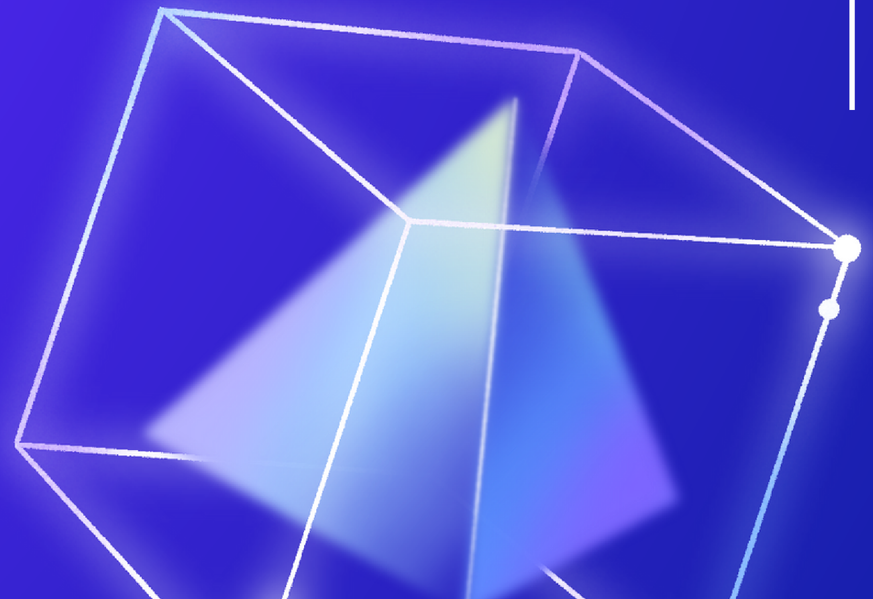
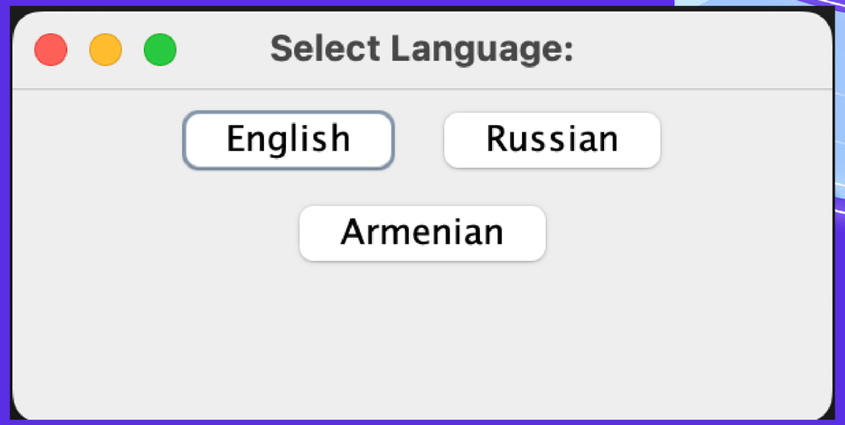


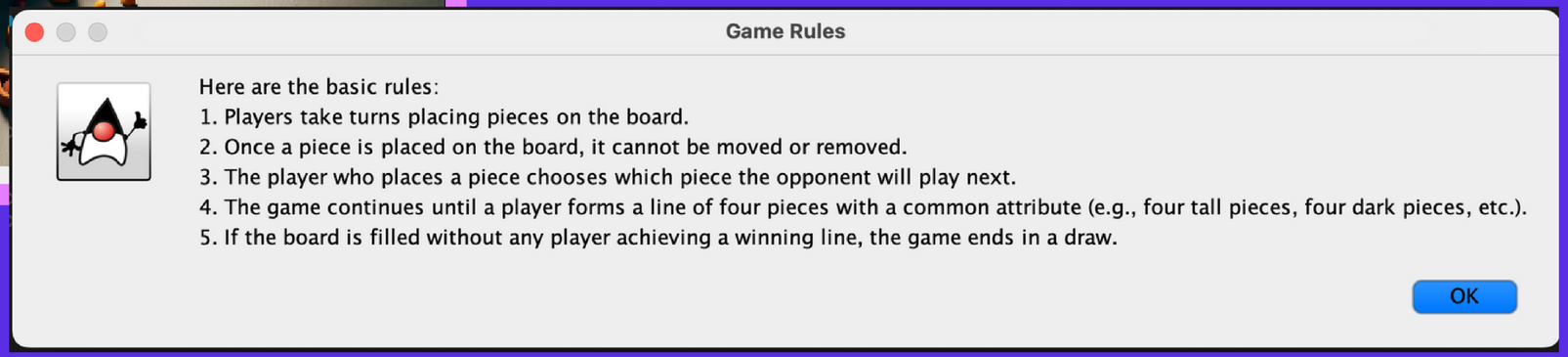
TABLE OF CONTENTS

- Introduction
- Project Structure
- Results and Demo game
- QA



INTRODUCTION





```
ast login: Thu May  9 16:36:51 on ttys060
achzak@Khachiks-MacBook-Pro ~ %
```

PROJECT STRUCTURE

CS120-IntroToOOP-Project Public

Pin | Unwatch 1 | Fork 0 | Starred 1

master | 1 Branch | 0 Tags | Go to file t | Add file | Code

zakaryan2107 remove gitignore rule — 32ee2a0 · 19 hours ago — 62 Commits

src	improving ui	yesterday
.gitignore	remove gitignore rule	19 hours ago
README.md	Adding multilanguage rules selection	2 days ago

README

Quarto Game

Welcome to Quarto Game, a Java implementation of the classic board game Quarto.

About

Quarto is a two-player abstract strategy game where players take turns placing pieces on a 4x4 board. The goal is to be the first to create a line of four pieces with a common attribute (color, shape, height, or fullness).

This project provides a graphical user interface (GUI) for playing Quarto against another human player. It features intuitive controls, visual representation of game pieces, and win/draw detection.

Features

About

No description, website, or topics provided.

Readme
Activity
1 star
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

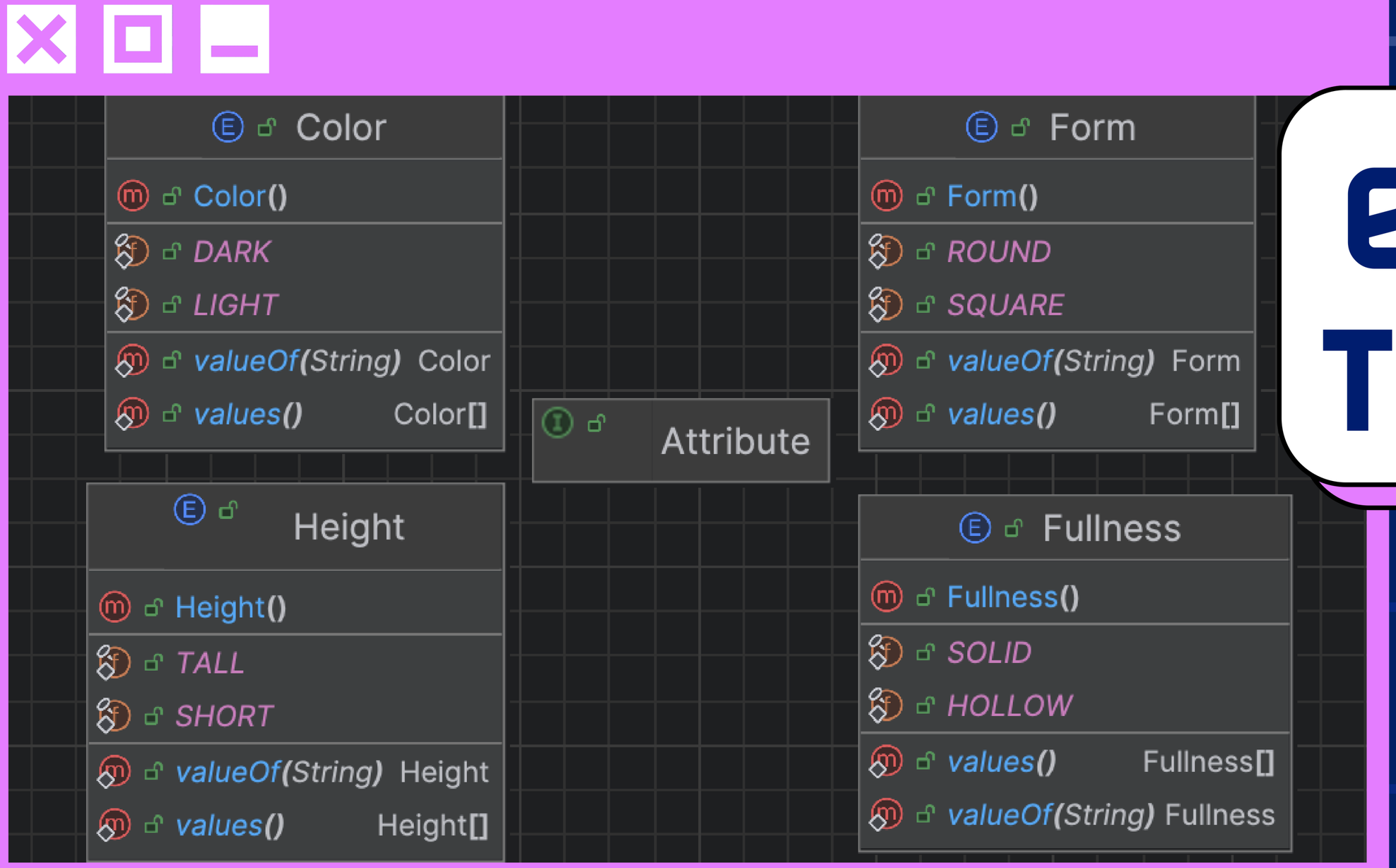
Languages

Java 100.0%

Suggested workflows

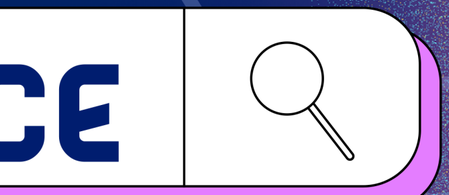
Based on your tech stack

- src
  - console
    - QuartoGame
  - core
    - Attributes
      - Attribute
      - Color
      - Form
      - Fullness
      - Height
    - Pieces
      - Piece
      - Pieces
    - Game
  - ui
    - PieceShapeButton
    - QuartoGUI
  - Main


Color
Color()
DARK
LIGHT
valueOf(String) Color
values() Color[]
Attribute
Form
Form()
ROUND
SQUARE
valueOf(String) Form
values() Form[]
Height
Height()
TALL
SHORT
valueOf(String) Height
values() Height[]
Fullness
Fullness()
SOLID
HOLLOW
values() Fullness[]
valueOf(String) Fullness

ENUM TYPES

INTERFACE

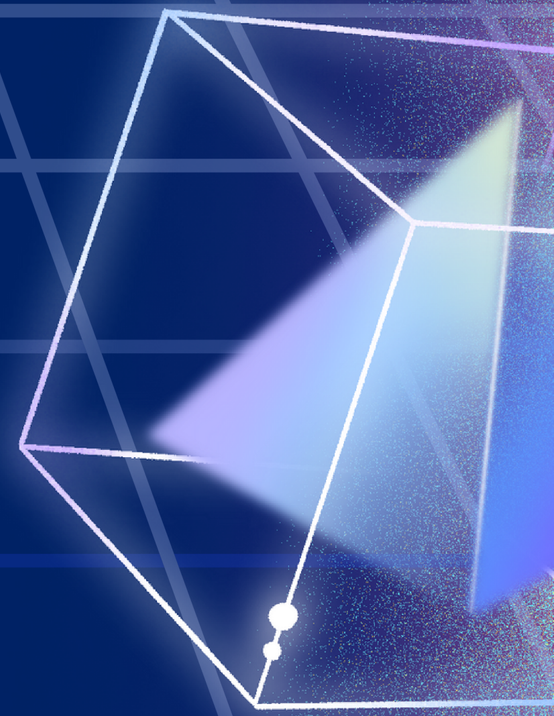


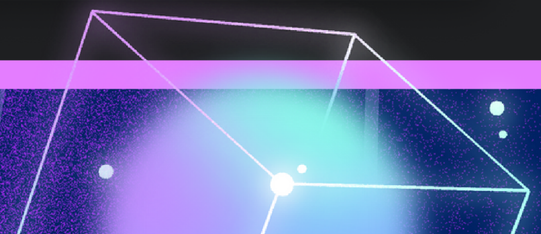
```java
/**
 * The Attribute interface represents a generic attribute for Quarto game pieces.
 * Implementing classes define specific attributes such as color, size, shape, etc.
 */
public interface Attribute {} 6 usages 4 implementations zakaryan2107
```

Interface core.Attributes.Attribute of core.Attributes 6 usages

Project Files Usages

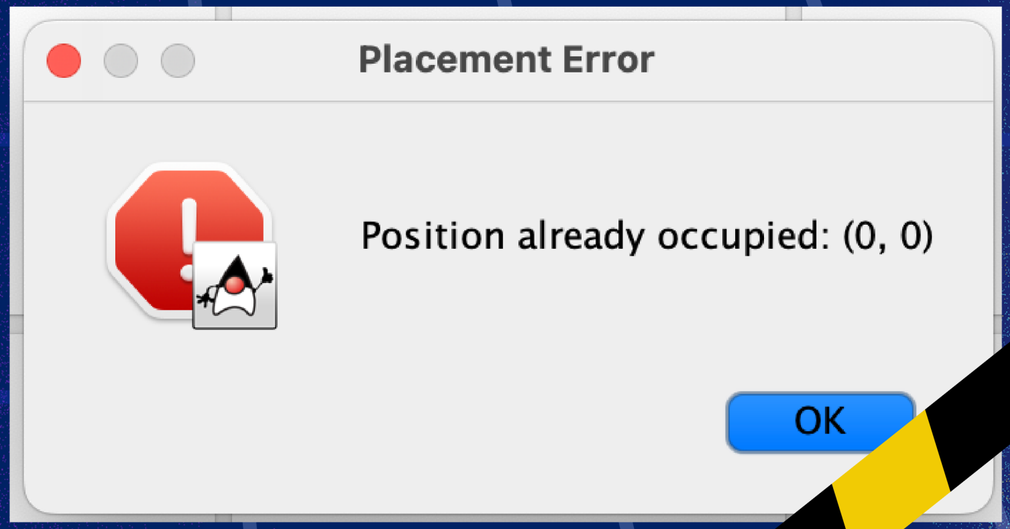
Color.java	7	public enum Color implements **Attribute** {
Form.java	7	public enum Form implements **Attribute** {
Fullness.java	7	public enum Fullness implements **Attribute** {
Height.java	7	public enum Height implements **Attribute** {
Game.java	183	private <TAttribute extends **Attribute**> boolean alll
Game.java	192	private <TAttribute extends **Attribute**> boolean isE



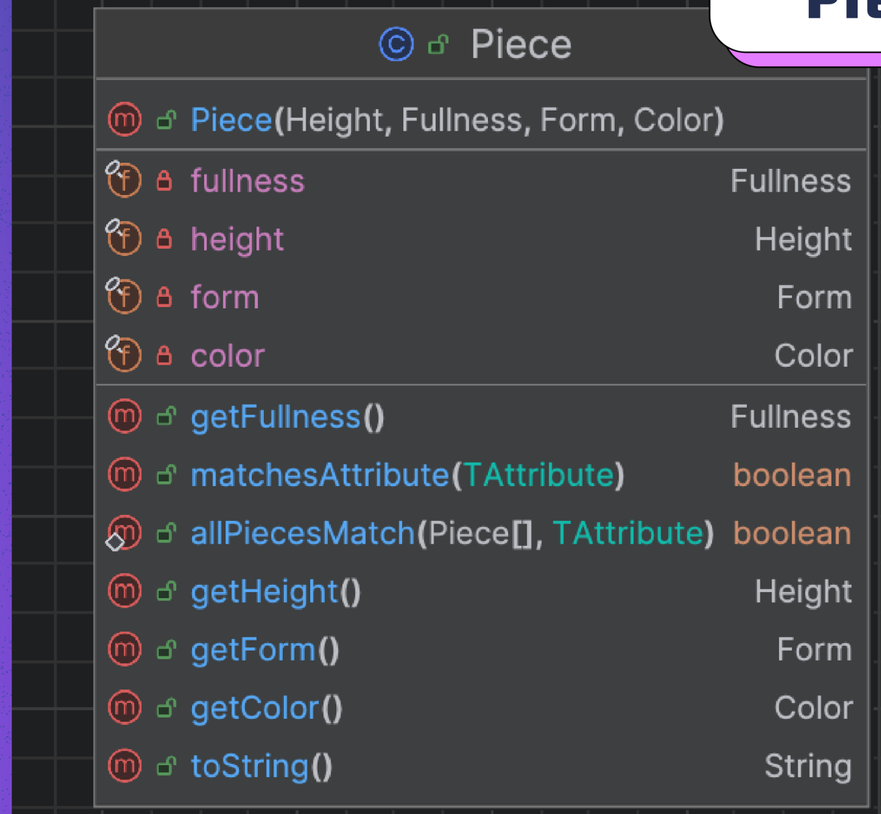


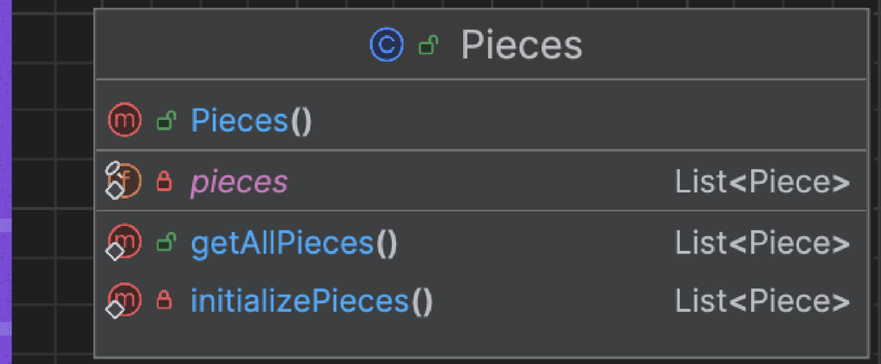
EXCEPTION HANDLING

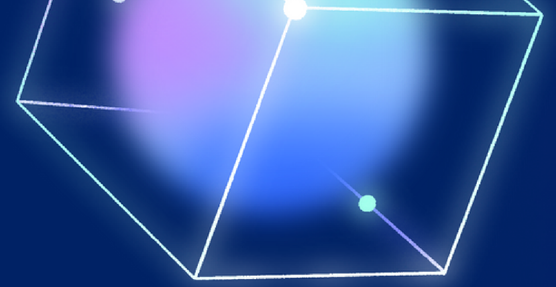
```java
package core.exceptions;

public class InvalidPiecePlacementException extends Exception {

    public InvalidPiecePlacementException(String message) {
        super(message);
    }
}
```



Piece Class





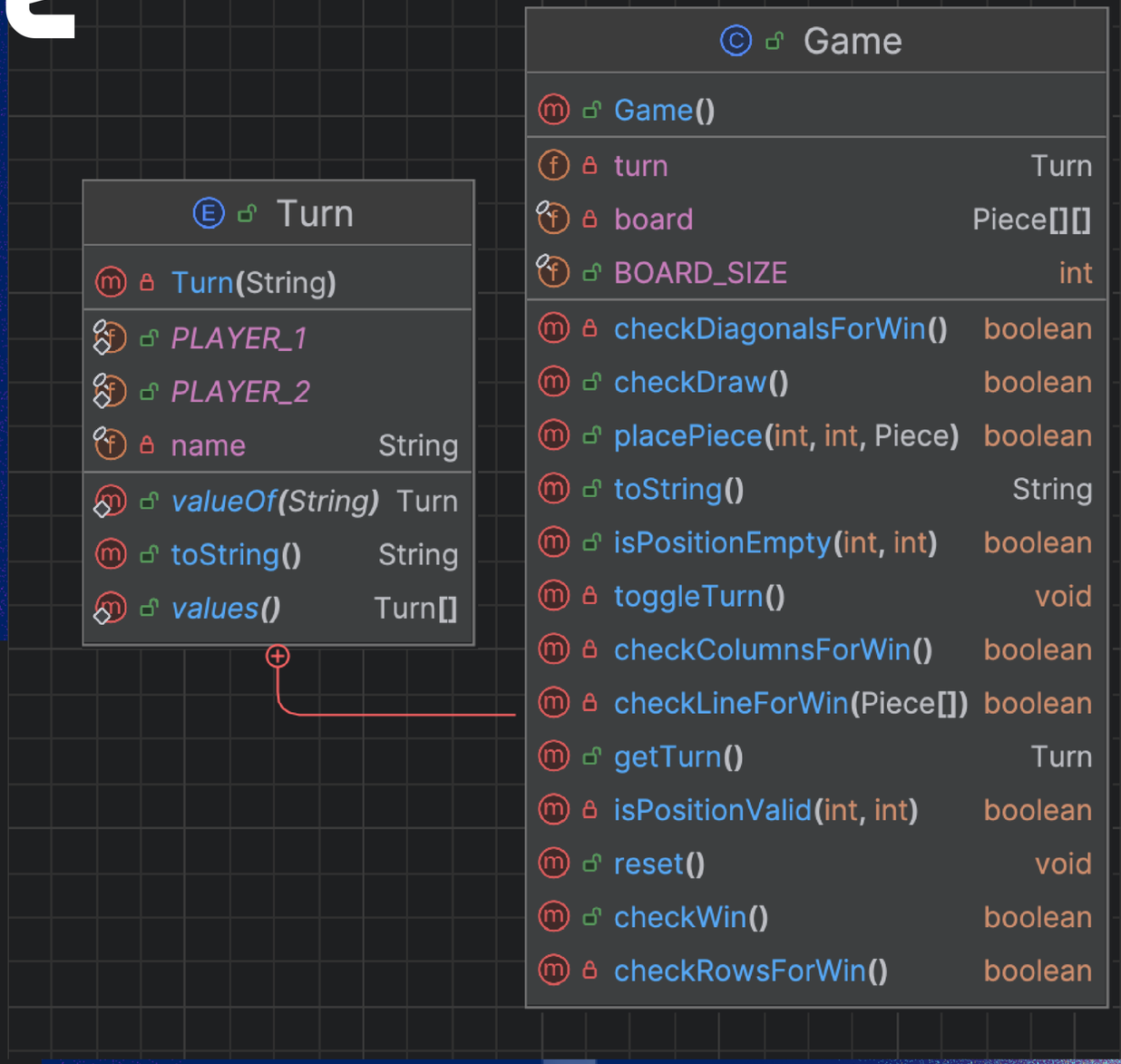


Many Pieces

```java
    /**
     * List to store all possible game pieces
     */
    private static final List<Piece> pieces = initializePieces();  1 usage

    /**
     * Initializes the list of game pieces
     * @return The list of generated game pieces
     */
    private static List<Piece> initializePieces() {  1 usage  Hrachya Hovakimyan
        List<Piece> pieces = new ArrayList<>();
        for (Height height : Height.values()) {
            for (Fullness fullness : Fullness.values()) {
                for (Color color : Color.values()) {
                    for (Form form : Form.values()) {
                        pieces.add(new Piece(height, fullness, form, color) {});  Hrachya Hovakimyan
                    }
                }
            }
        }
        return pieces;
    }
}
```

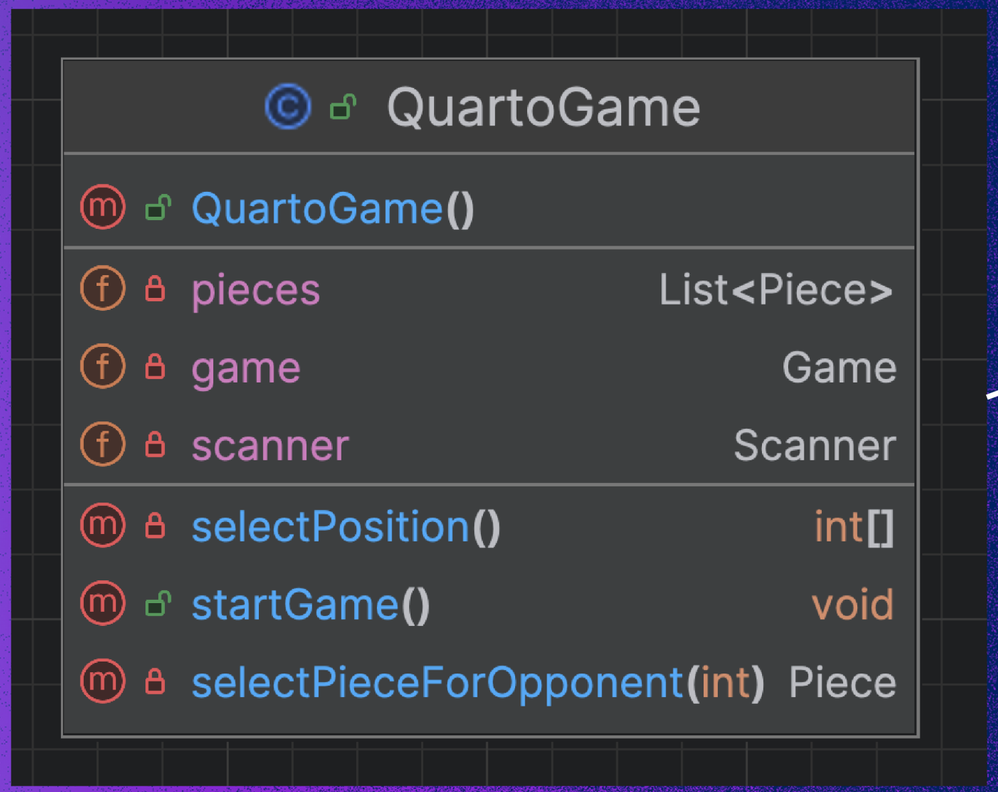
WHAT ABOUT THE GAME LOGIC?

Our vision



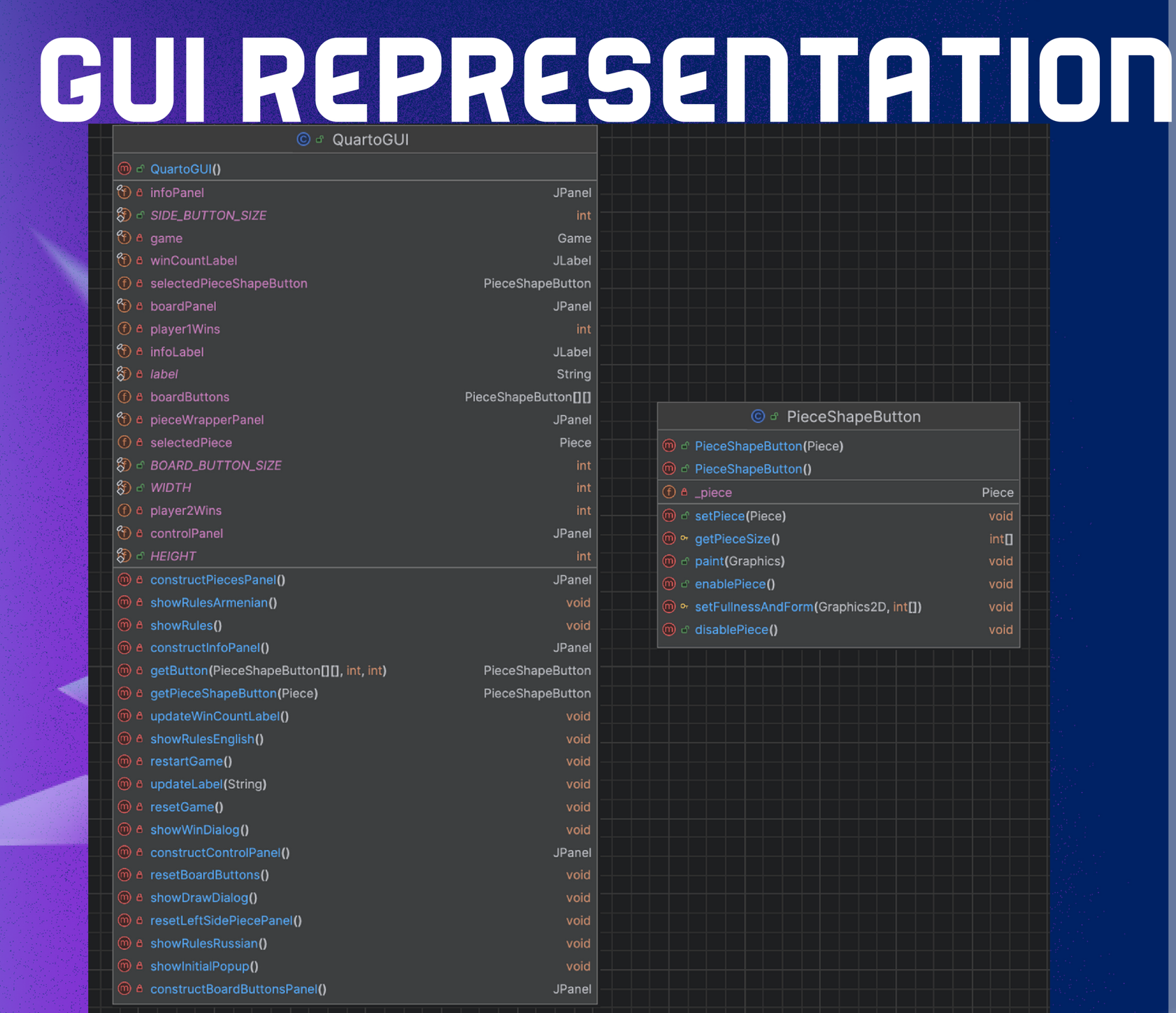

```java
/**
 * Resets the game by clearing the game board and setting the turn to Player 1.
 */
public void reset() {  1 usage  zakaryan2107
    for (int i = 0; i < BOARD_SIZE; i++) {
        for (int j = 0; j < BOARD_SIZE; j++) {
            board[i][j] = null;
        }
    }
    // Reset turn to PLAYER_1
    turn = Turn.PLAYER_1;
}
```

"WHITE AND BLACK" GAME REPRESENTION

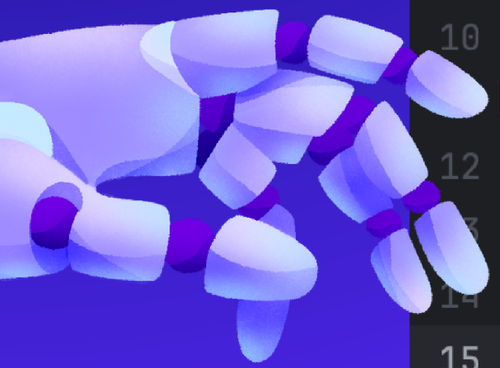


```java
public void startGame() throws InvalidPiecePlacementException{  no usages
    System.out.println("Welcome to Quarto!");
    System.out.println("Let's start the game.");

    while (!game.checkWin() && !pieces.isEmpty()) {
        // Player 1's turn
        System.out.println(game); // Print the current game board
        System.out.println("Player 1's turn:");
        Piece pieceForOpponent = selectPieceForOpponent(1);
        int[] position = selectPosition();
        game.placePiece(position[0], position[1], pieceForOpponent);
        if (game.checkWin()) {
            System.out.println("Player 1 wins!");
            break;
        }
        if (pieces.isEmpty()) {
            System.out.println("No more pieces to select. It's a draw!");
            break;
        }

        // Player 2's turn
        System.out.println(game); // Print the current game board
        System.out.println("Player 2's turn:");
        pieceForOpponent = selectPieceForOpponent(2);
        position = selectPosition();
        game.placePiece(position[0], position[1], pieceForOpponent);
        if (game.checkWin()) {
            System.out.println("Player 2 wins!");
            break;
        }
    }

    // Check for draw after the loop ends
    if (!game.checkWin() && pieces.isEmpty()) {
        System.out.println("No more pieces to select. It's a draw!");
    }
}
```

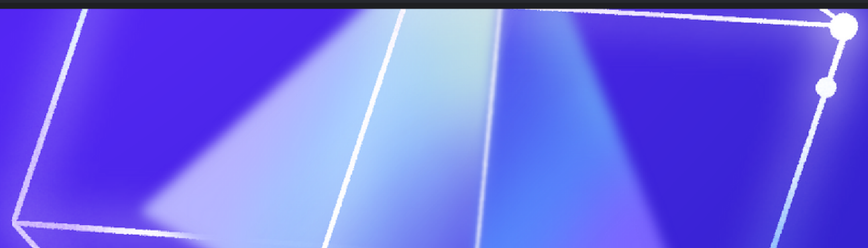
GUI REPRESENTATION



THE "MAIN" PART...



```java
import javax.swing.SwingUtilities;
import ui.QuartoGUI;
/**
 * Main Class for execution
 */
public class Main {  zakaryan2107
    /**
     * Execute the QuartoGUI constructor within the event dispatch thread
     * @param args Argument (Not used)
     */
    public static void main(String[] args) {  zakaryan2107
        SwingUtilities.invokeLater(QuartoGUI::new);
    }
}

```



```
public class Main {
    public static void main (String[] args) {
        System.out.print(“Thank you!”);
    }
}
```