Progetto di Programmazione 3 e Laboratorio

Gioco Forza 4 (IA)

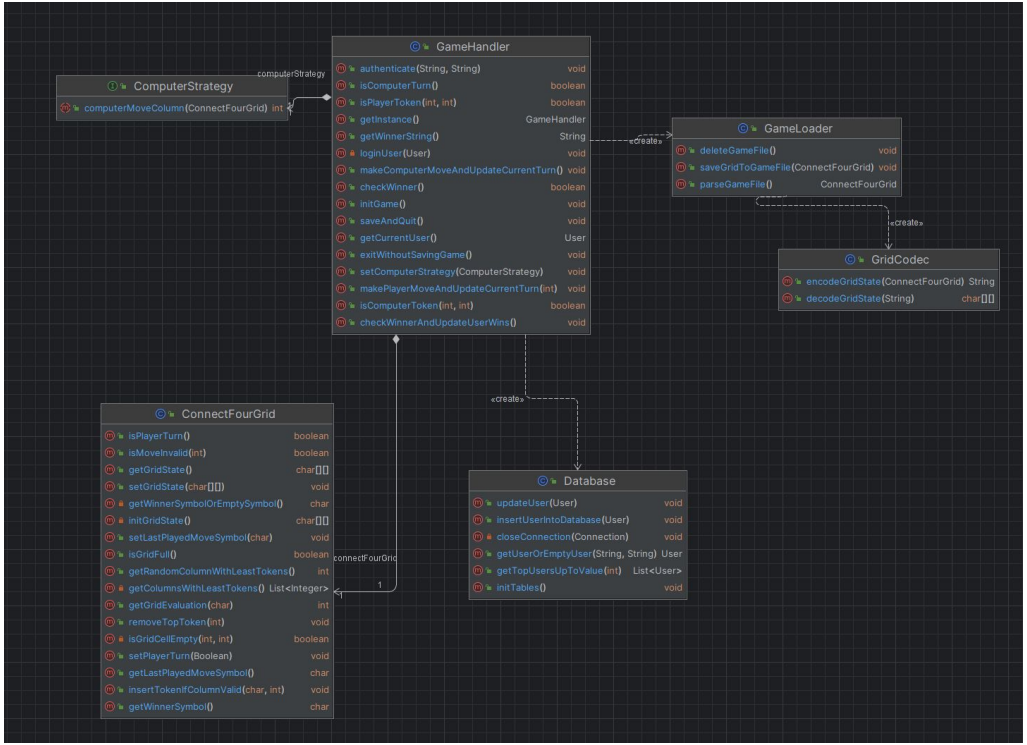
Anno Accademico
2023-2024

Candidato

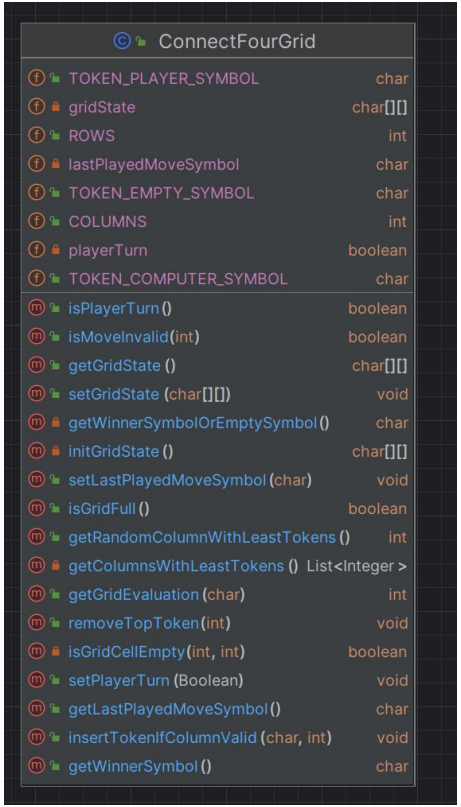
Raffaele Talente
0124002658

Requisiti

- ➔ Programma che permetta di simulare il gioco Forza 4
- ➔ Utente VS Computer
- ➔ Il computer deve inserire le pedine in base a delle modalità di gioco specifiche selezionabili dall’utente
- ➔ Nome, cognome e numero di vittorie di un utente devono essere memorizzati
- ➔ Deve essere possibile sospendere una partita e riprenderla successivamente

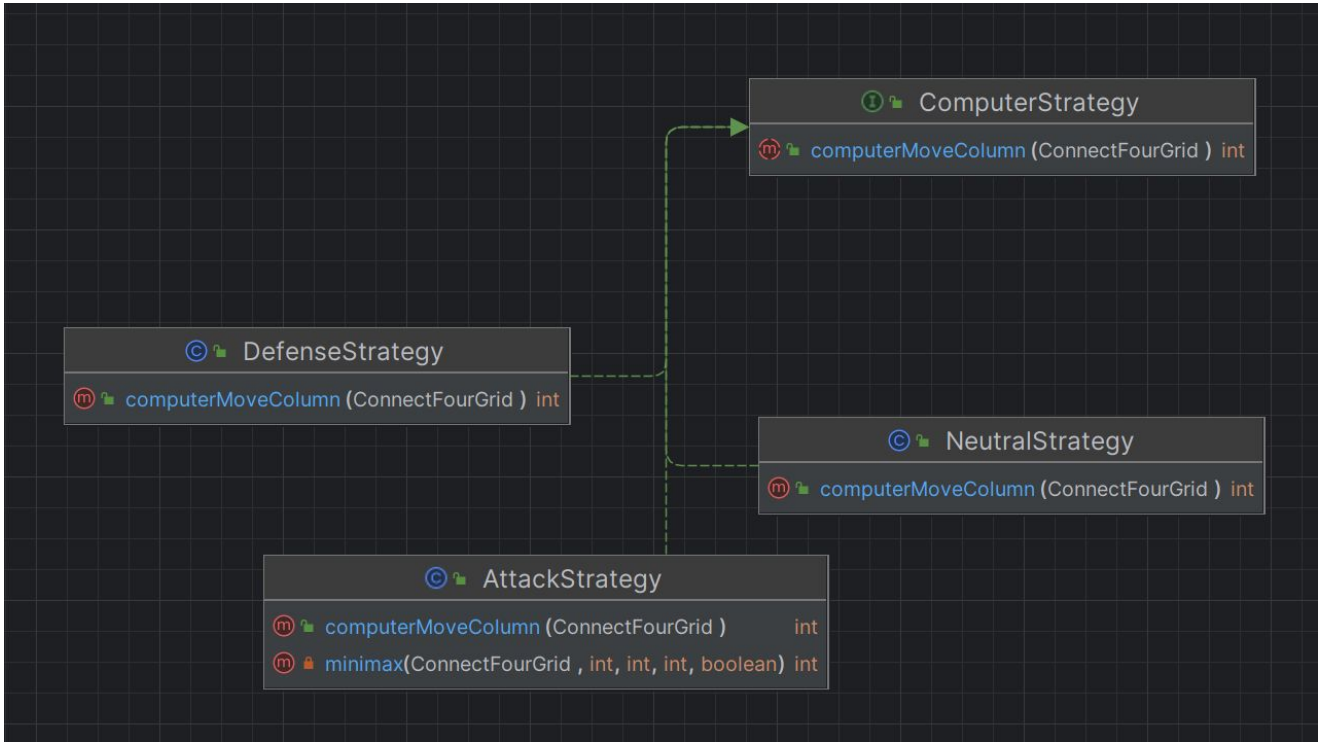


La classe GameHandler gestisce il flusso della partita



La classe ConnectFourGrid incapsula il comportamento della griglia di gioco

ComputerStrategy:
Il Design Pattern Strategy



DefenseStrategy

```
    public int computerMoveColumn(ConnectFourGrid connectFourGrid) {
    char symbolToCount = connectFourGrid.TOKEN_PLAYER_SYMBOL;
    Map<String, Integer> playableColumnsWithMostConnectedTokens
                          = new HashMap<>();
    // horizontal
    for(int i = 0; i < connectFourGrid.ROWS; i++) {
        for(int j = 0; j < connectFourGrid.COLUMNS - 3; j++) {
            if(connectFourGrid.isMoveInvalid(j)) {
                continue;
            }
            int currentCount = 0;
            for(int k = 0; k <= 3; k++) {
                if(connectFourGrid.getGridState()[i][j + k]
                        == symbolToCount) {
                    currentCount++;
                }
            }
            playableColumnsWithMostConnectedTokens
                .put("h"+j, currentCount);
        }
    }
    // stessa operazione per le altre direzioni
    // ...
    // codice omesso per semplicità

    String highestKey = Collections.max(
            playableColumnsWithMostConnectedTokens.entrySet(),
            Map.Entry.comparingByValue()).getKey();

    int value = playableColumnsWithMostConnectedTokens
                    .get(highestKey);
    // regex per togliere i caratteri non-cifre
    int column = Integer.parseInt(highestKey
                        .replaceAll("[^\\d.]", ""));
    if(value == 0 ||
       value == 1 ||
        connectFourGrid.isMoveInvalid(column)) {
        return connectFourGrid.getRandomColumnWithLeastTokens();
    }
    return column;
}
```

AttackStrategy

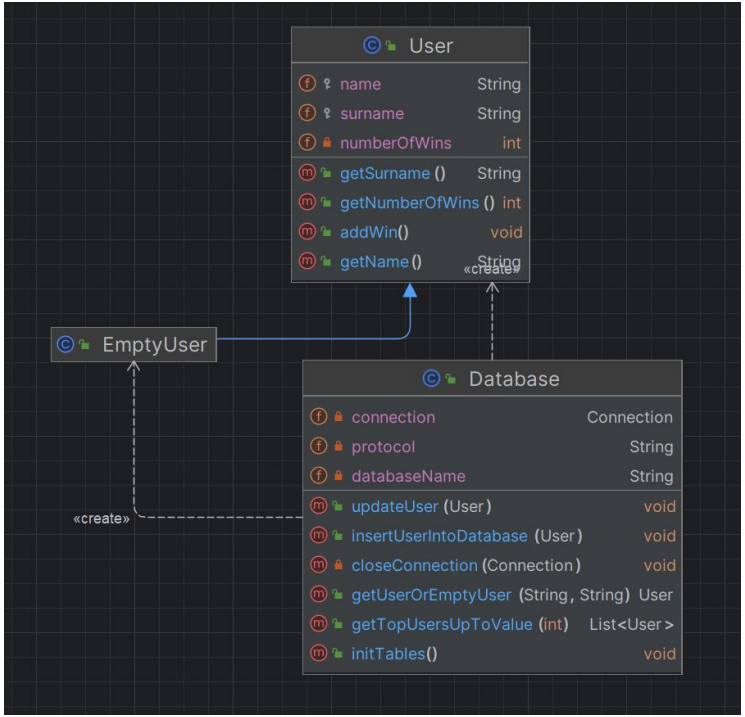
```
    private int minimax(ConnectFourGrid connectFourGrid, int depth,
                        int alpha, int beta, boolean isComputerTurn) {
    if(depth <= 0) {
        return 0;
    }
    int currentGridEvaluation = connectFourGrid.getGridEvaluation(
                                connectFourGrid.TOKEN_COMPUTER_SYMBOL);
    if(currentGridEvaluation == 1) {
        return depth;
    } else if(currentGridEvaluation == -1) {
        return -depth;
    } else if(connectFourGrid.isGridFull()) {
        return 0;
    }

    if(isComputerTurn) {
        int maxValue = Integer.MIN_VALUE;
        for(int j = 0; j < connectFourGrid.COLUMNS; j++) {
            if(connectFourGrid.isMoveInvalid(j)) { continue; }
            connectFourGrid.insertTokenIfColumnValid(
                            connectFourGrid.TOKEN_COMPUTER_SYMBOL, j);
            maxValue = Math.max(maxValue,
                        minimax(connectFourGrid, depth - 1,
                                        alpha, beta, false));
            connectFourGrid.removeTopToken(j);
            alpha = Math.max(alpha, maxValue);
            if(alpha >= beta) {
               break;
            }
        }
        return maxValue;
    } else {
        int minValue = Integer.MAX_VALUE;
        for(int j = 0; j < connectFourGrid.COLUMNS; j++) {
            if(connectFourGrid.isMoveInvalid(j)) { continue; }
            connectFourGrid.insertTokenIfColumnValid(
                            connectFourGrid.TOKEN_PLAYER_SYMBOL, j);
            minValue = Math.min(minValue,
                        minimax(connectFourGrid, depth - 1,
                                        alpha, beta, true));
            connectFourGrid.removeTopToken(j);
            beta = Math.min(beta, minValue);
            if(alpha >= beta) {
                break;
            }
        }
        return minValue;
    }
}
```

Minimax con e senza Alpha-beta pruning

Profondità	Minimax	Alpha-beta
2	0.9 ms	0.9 ms
3	7.0 ms	2.5 ms
4	31.1 ms	8.1 ms
5	56.4 ms	19.8 ms
6	109.6 ms	31.7 ms
7	341.2 ms	44.1 ms
8	1.5 s	53.2 ms
9	10.4 s	100.0 ms
10	1 min	159.3 ms
11	7.4 min	387 ms
12		1.2 s
13		6.1 s
14		21.2 s
15		2 min
16		6.5 min

NeutralStrategy

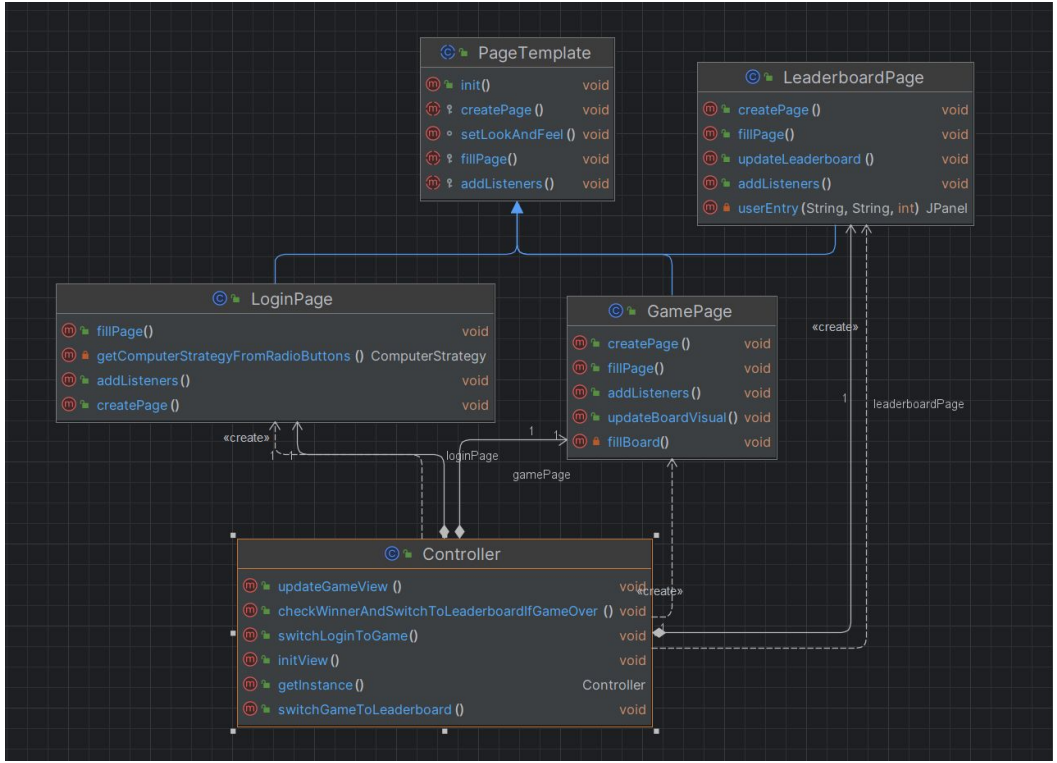
```
        public int computerMoveColumn(ConnectFourGrid connectFourGrid) {
    Random r = new Random();
    boolean coin = r.nextBoolean();
    if(coin) {
        return (new AttackStrategy()).computerMoveColumn(connectFourGrid);
    } else {
        return (new DefenseStrategy()).computerMoveColumn(connectFourGrid);
    }
}
```


User
name String
surname String
numberOfWins int
getSurname () String
getNumberOfWins () int
addWin() void
getName () String
«create»
EmptyUser
Database
connection Connection
protocol String
databaseName String
updateUser (User) void
insertUserIntoDatabase (User) void
closeConnection (Connection) void
getUserOrEmptyUser (String, String) User
getTopUsersUpToValue (int) List<User>
initTables() void
«create»

II Database

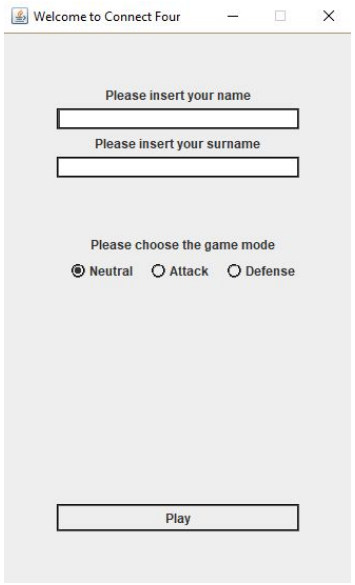

SQLite

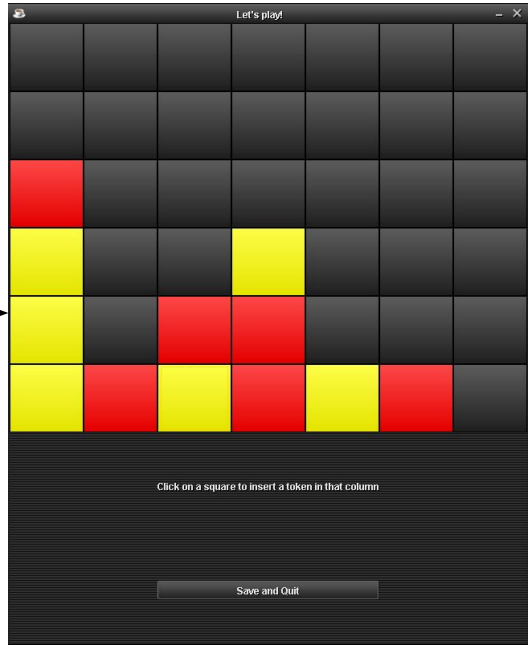
Interfaccia Grafica con Swing



PageTemplate:
Il Design Pattern Template Method

```
public abstract class PageTemplate extends JFrame {
    public final void init() {
        createPage();
        fillPage();
        setLookAndFeel();
        addListeners();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
    }

    protected abstract void createPage();

    // metodo di hook
    protected void fillPage() {
    }

    // metodo final:
    // tutte le pagine devono avere lo stesso look and feel
    final void setLookAndFeel() {
        try {
            UIManager.setLookAndFeel(
                    "com.jtattoo.plaf.hifi.HiFiLookAndFeel
                                        ");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    protected abstract void addListeners();
}
```


Welcome to Connect Four
Please insert your name
Please insert your surname
Please choose the game mode
Neutral
Attack
Defense
Play


Let's play!
Click on a square to insert a token in that column
Save and Quit


You win!
Leaderboard
programmazione tre 3
raffaele talente 1
mario rossi 0
maria rossi 0
Quit