|  |  |  |
| --- | --- | --- |
| **Shokhrukh Nigmatillaev** | **2. Assignment / Task 4** | 11.12.2024 |
| APVAVZ | [apvavz@inf.elte.hu](mailto:apvavz@inf.elte.hu) | Group 5 |
| **Task** |  |  |

**Tricky-five-in-a-row**  
Create a game, which is a variant of the well-known five-in-a-row game. The two players can play on a board consists of n x n fields. Players put their signs alternately (X and O) on the board. A sign can be put only onto a free field. The game ends, when the board is full, or a player won by having five adjacent signs in a row, column or diagonal. The program should show during the game who turns.

The trick in this variant is that if a player makes 3 adjacent signs (in a row, column or diagonal), then one of his signs is removed randomly (not necessary from this 3 signs). Similar happens, when the player makes 4 adjacent signs, but in this case two of his signs are removed.

Implement this game, and let the board size be selectable (6x6, 10x10, 14x14). The game should recognize if it is ended, and it has to show in a message box which player won (if the game is not ended with draw), and automatically begin a new game.

# Summary:

The Tricky five in one is created as a version of a popular Five in one game, but the little bit of trick that if you just continue putting your signs straight, you will not win. So, we have 3 boards and each time after the game you have the possibility to restart it. X player is represented as green and O player represented as red. Player who gets 5 in row first wins the game.

**Analysis:**

For the given game, I would like to implement Java GUI. First it will open welcome page where will be only one button and this will be separate OpeningPage class. Then I would like to implement ChooseGameBoard class. For view there is one abstract class called GameField and 3 children classes that inherit from it. For model there is GameLogic class that contains full game logic, with the winner, draw, removal of the dots etc.

UML:  
A diagram of a computer program

Description automatically generated with medium confidence  
**Methods :**

**Main.java**  
- main: Initializes and displays the main game window.

# OpeningPage.java

# -createLabel, createButton – they both create main components of the main page Label and Button

-createFrame – Frame packs Button and Label

**ChooseGamePage.java**  
-createButtons – we create 3 buttons on the page and each of them will be responsible for it’s board size

-createFrame – creates frame and fixes the buttons on it

**Abstract GameField.java**  
+GameField(int size) - Initializes the game board of a given size, setting up the frame, board, and top panel with player information.

# -createFrame - Creates frame for the game

# -createBoard - Initializes a grid of buttons (JButton) to represent the game board.

# -createTopBoard - Creates the top panel

# -navigateBack : Closes the current game and navigates back to the game choosing page

# GameLogic.java:

# -handlePlayerMove - Validates and processes a player's move:

# Win and Draw Conditions

# -checkWin - Determines if a player has won by checking for 5 consecutive symbols in any direction.

# -checkDirection - Evaluates a specific direction for consecutive player symbols.

# -countConsecutive - Counts consecutive symbols starting from a given cell in a specified direction

# -showEndGameMessage - Displays the end-game message for a win or draw.

# -getMaxAdjacentCount - Calculates the maximum consecutive symbols for a player across all directions.

# -removeRandomSigns - Randomly removes a specified number of the player's symbols from the board

# EasyField.java - +EasyField – creates easy board MediumField.java

# +MediumField – creates easy board

# HardField.java

# +HardField – creates easy board

# Running application:

# A screenshot of a computer Description automatically generated A screenshot of a computer A screenshot of a game Description automatically generated A screenshot of a game Description automatically generatedA screenshot of a gameA screenshot of a game A screenshot of a computer screen Description automatically generated

**Testing:**

**White Box Testing**

White box testing involves looking at the internal structure of code

1. Game starts  
   **AS** A user  
   **I WANT** TO start the game  
   **GIVEN** I am on the **ChooseGamePage**  
   **WHEN** I select a board size (e.g., 6x6, 10x10, 14x14)  
   **THEN** the corresponding game board is displayed, and the game begins
2. **Invalid Move on Occupied Cell**  
   **AS** A player   
   **I** **WANT** TO be restricted from making an invalid move  
   **GIVEN** I have selected an already occupied cell  
   **WHEN** I try to make a move in that cell  
   **THEN** the game should prevent the move and keep the current player
3. **Win with horizontal alignment**  
   **AS** A player  
   **I WANT** TO win by aligning 5 symbols horizontally  
   **GIVEN** there are 4 of my symbols already aligned horizontally  
   **WHEN** I place my fifth symbol in the consequent cell  
   **THEN** I win, and the victory message is displayed.
4. **Win with vertical alignment**  
   **AS** A player  
   **I** **WANT** TO win by aligning 5 symbols vertically  
   **GIVEN** there are 4 of my symbols already aligned vertically  
   **WHEN** I place my fifth symbol in the consequent cell  
   **THEN** I win, and the victory message is displayed.
5. **Win with Diagonal alignment from top-left to bottom-right  
   AS A player**  
   I WANT TO win by aligning 5 symbols diagonally from top-left to bottom-right  
   **GIVEN** there are 4 symbols aligned diagonally in that direction  
   **WHEN** I place the fifth symbol in the correct cell  
   **THEN** I win, and a victory message is displayed
6. **Win with Diagonal from Top-Right to Bottom-Left**  
   **AS A player**  
   I WANT TO win by aligning 5 symbols diagonally from top-right to bottom-left  
   **GIVEN** there are 4 symbols aligned diagonally in that direction  
   **WHEN** I place the fifth symbol in the correct cell  
   **THEN** I win, and a victory message is displayed
7. **Draw** **case**  
   **AS A player  
   I WANT TO** see if draw result is available  
   **GIVEN** the board is full and none of the players have 5 in row same symbols  
   **WHEN** I place the last sign  
   **THEN** the game ends with the draw and message pops up informing about it
8. **Remove 1 Symbol for 3 Consecutive Symbols  
   AS A player  
   I WANT TO** have one of my symbols removed **GIVEN** I align 3 consecutive symbols **WHEN** I complete the third consecutive symbol **THEN** one of my symbols is randomly removed (not specifically from these 3 in a row)
9. **Remove 2 Symbols for 4 Consecutive Symbols  
   AS A player  
   I WANT TO** have two of my symbols removed  
   **GIVEN** I align 4 consecutive symbols **WHEN** I complete the fourth consecutive symbol **THEN** two of my symbols are randomly removed (not specifically from these 4 in a row)
10. **Restart Game After Win/Lose  
    AS A user  
    I WANT TO** restart the game **GIVEN** I have just won/lost the game **WHEN** I choose to restart **THEN** a new game board is initialized, and I can play again
11. **Restart Game After Draw  
    AS A** user **I WANT TO** restart the game **GIVEN** the game ends in a draw **WHEN I** choose to restart  
    **THEN** I will be taken to the page where I can choose the game board and start again
12. **Navigate Back to Game Selection  
    AS A** user **I WANT TO** return to the game selection menu **GIVEN** I am on the game board **WHEN** I click the "Back" button **THEN** I am returned to the ChooseGamePage
13. **Fast Consecutive Moves  
    AS A** player **I WANT** TO ensure fast moves do not break the game **GIVEN** I make consecutive moves rapidly on a valid game board **WHEN** I click quickly between turns **THEN** the game correctly updates the board and switches players without errors
14. **Invalid Move After Game End  
    AS A** player **I WANT TO** be restricted from making moves after the game ends **GIVEN** the game has ended with a win or draw **WHEN** I try to click on a cell **THEN** the game ignores the move, and the board remains unchanged
15. **Rules of each board are the same  
    AS A** player  
    **I WANT TO** experience game in each board  
    **GIVEN** the game has rules and tricky parts  
    **WHEN** I open any of the board  
    **THEN** gameplay will be the same and experience will be the same