DET 205 Object-oriented Design and Programming

Project Report Form

|  |  |
| --- | --- |
| Student ID | S24214373 |
| Student Name | Ho Nok Hei (Hugo) |
| Topic | |
| 369 Broad Game | |
| Project Description | |
| This program is a strategy game played on a 9x9 board where players take turns placing pieces on empty spaces, aiming to form sequences of pieces in multiples of 3 to score points.  The scoring rules are based on the number of consecutive pieces. When a player forms a sequence of 3, 6, or 9 consecutive pieces, they earn the corresponding points. The game supports human vs. AI gameplay, with three difficulty levels: Easy, Medium, and Hard. | |
| Algorithm Design | |
| 1. Board Representation: A 9x9 two-dimensional array is used to represent the board, where 0 indicates an empty space, 1 represents Player 1, and 2 represents Player 2. 2. Score Calculation: After each move, the game checks for consecutive pieces in four directions (left, up, upper-left, lower-left), counts the number of consecutive pieces, and calculates the score based on whether the count is a multiple of 3. 3. AI：   Easy Mode: The AI randomly selects an empty space to place a piece and delays for 1 second to simulate thinking.  Medium Mode: The AI selects the position that can yield the highest score. If no scoring opportunity exists, it randomly selects a position.  Hard Mode: The AI considers the opponent's potential score, weights its own score against the opponent's, and selects the optimal position. The weighting dynamically adjusts based on the number of empty spaces on the board.   1. Game Flow Control: Players take turns placing pieces. After each move, the game checks if it has ended and determines the winner based on the scores. 2. This program has two versions: Console version and GUI version. Each version has 3 modes: Play with AI, Play with Friend, Watch AI Battle 3. We also have two languages you can select: zh\_TW and en\_US | |
| User Manual | |
| 1. Start the Game: Run ` run\_program.bat` and choose between the console version or the GUI version and modes. 2. Make a Move: In the console version, input coordinates (e.g., A1) to place a piece. In the GUI version, click on an empty space on the board to place a piece. 3. AI Battle: Select the AI difficulty level to play against the AI. 4. Game Over: The game ends when the board is full. The winner is then displayed. | |
| Sample Run | |
| Select languages:    Select version: | |
| Console version:   * Select mode: * Play with AI: * Play with Friend: * Watch AI Battle: | |
| GUI version:   * Select mode:  * Play with AI:  * Play with Friend: * Watch AI Battle: | |
| Job Division | |
| Me:   1. Board Logic: Responsible for board initialization, score calculation, and game-over determination. 2. AI Implementation: Implemented the decision-making logic for the three AI difficulty levels. 3. User Interface: Developed the console and GUI versions of the user interface to provide an intuitive user experience. 4. Testing and Debugging: Tested all game features to ensure stable operation. | |
| Reference | |
| 1. Java Swing Official Documentation：<https://docs.oracle.com/javase/tutorial/uiswing/> | |