Testing strategy and Testing cases

Testing Strategy:

Our testing strategy follows a **combination of unit testing and integration testing** using **QTest framework** provided by Qt. This ensures that individual components function correctly and that the interactions between them work as expected. The strategy is structured as follows:

Unit Testing:

We perform unit tests to validate the core logic of the GameBoard class:

- **Valid move logic**: Ensuring valid moves are accepted and placed on the board.
- **Invalid move rejection**: Preventing moves on already occupied cells.
- **Winner detection**: Verifying that winners are correctly identified across rows, columns, and diagonals.
- **Board state checking**: Detecting when the board is full.

Integration Testing:

We also test how different classes interact to ensure the system behaves as expected when components are combined:

- **AI Integration**: Verifying that switching players in PvAI mode triggers the AI to make a move.
- **Replay Integration**: Ensuring that clicking the replay button in the UI loads previously saved game records into the dropdown menu.
- **Turn Management Integration**

  Tests that the game correctly alternates between players and processes each move.

Test Cases:

Unit Testing:

1. **testMakeMoveValid()**
   Verifies that a valid move is accepted and stored on the board.
2. **testMakeMoveInvalid()**
   Tries to make a move on an already occupied cell and checks that it's rejected.
3. **testCheckWinnerRows()**
   Checks if a player is detected as winner when they occupy an entire row.
4. **testCheckWinnerColumns()**
   Verifies win detection when a column is filled by the same player.
5. **testCheckWinnerDiagonals()**
   Checks diagonal win condition (from top-left to bottom-right).

Integration Testing:

1. **testFullTurnCycle()**
   Simulates turns between two players, verifying correct switching and game state after each move.
2. **testAIIntegration()**
   Makes a human move, then ensures the AI responds (after delay), and checks that control returns to the human.
3. **testReplayIntegration()**
   Loads a previously saved game into the UI and verifies that it appears in the combo box for replay.