CHESS: BATTLE OF THE QUEENS

Description:

Upon launching the game, you'll encounter five options displayed on your screen:

1. Play
2. Game Mode
3. Game Replays
4. Settings
5. Exit

Play: Selecting 'Play' will present you with the following five choices:

- VS Player: This option initiates a classic chess game against another player.
- VS Computer: Embark on a classic game against a computer opponent. The difficulty level for this mode can be adjusted in the Settings menu.
- Load MVC.PGN Game: This option allows you to load a MVC.PGN game against a player and resume play.
- Situation Builder: This feature lets you construct any chess scenario and play it out.
- Back: This option navigates back to the main menu.

Game Mode: This choice initiates a chess game with special rules based on settings chosen in the Settings menu. Three modes are available:

- The first mode generates a random piece type each turn, and only this piece type can be moved.
- The second mode randomly selects a specific piece each turn, allowing only that piece to be moved.
- The third mode randomly alternates between the first two modes every turn.

Game Replays: This feature allows you to import a MVC.PGN game from the clipboard or select a MVC.PGN file to replay and analyze previous games.

Settings: Various customization options are available in this menu:

- GameMode Animation: Toggle the GameMode animations on or off.
- Dev Mode: This developer setting allows you to place pieces on the board using the right mouse button.
- Computer Difficulty: Adjust the difficulty of the computer opponent. Options include Random, Easy, and Cheater.
- GameMode: Choose one of the previously mentioned types of the GameMode.
- Clock: Toggle the in-game clock counter on or off.
- Time on Clock: Set the amount of time each player is allotted for the game.

Test Strategy:

Our testing strategy is primarily centered around the 'Play' functionality, with particular emphasis on the game engine and core mechanics of the player versus player (PvP) mode. Following that, we intend to rigorously test the 'Game Replay' and PGN-related features of the game.

Test Levels:

1. Unit Tests:
   Our unit tests are predominantly focused on individual chess pieces and the game board itself. We aim to thoroughly evaluate the following aspects:
   - Calculation of potential moves for each chess piece.
   - Placing pieces onto the game board.
   - Removing pieces from the game board.
   - Locating specific pieces on the game board.
   - Moving a piece from one position to another.
2. Integration/Process Tests:
   These tests mainly concentrate on the game engine, PGN features, and

the menu system. Specific scenarios include:

- Castling: We test the special chess move involving a player's king and one of their rooks.
- Checkmate: This involves verifying the correct detection of the 'checkmate' game state.
- En Passant: We test this unique pawn capturing move.
- Stalemate: This involves checking for accurate identification of a 'stalemate' situation.
- Promotion: We assess the proper implementation of pawn promotion.
- Loading PGNs: We evaluate the functionality to load games in Portable Game Notation (PGN) format.
- Saving PGNs: The game's ability to correctly save games in PGN format is tested.
- Navigation within the menu

Testing Scenarios:

In terms of testing inputs, we have elected to focus on two main areas:

- We are testing the 'config.txt' input file that allows players to determine the settings of the game.
- We are also assessing PGN files, which are a standardized format (Portable Game Notation) for recording and replaying chess games.

Equivalence classes:

Config:

Business:

- dev
  - valid options are only 0 or 1 (OFF or ON)
- computerDifficulty
  - valid options are only 0, 1 or 2 (EASY, NORMAL, CHEATER)
- gameMode
  - valid options are only 0 or 1 (RANDOM or CHAOS)
- clock
  - valid options are only 0 or 1 (OFF or ON)
- clockTime
  - valid options are only 0 or 1 (1 or 10 minutes)
- animation
  - valid options are only 0 or 1 (OFF or ON)

Technical:

- the keys in the config file has to be strings in that exact order as above
- there are no equivalence classes because in case of any invalid inputs, the game won’t start
- the values has to be exactly as mentioned above or the game won’t start
- the keys and values have to be separated by one space and the keys by a line break

dev	computerDifficulty	gameMode	clock	clockTime	animation
0	0	1	1	1	1
1	0	0	0	0	0
0	1	0	1	0	1
1	1	1	0	1	0
0	2	0	0	1	1
1	2	1	1	0	0
0	0	1	0	1	0
1	2	0	1	0	1
-1	0	1	1	1	1
2	0	0	0	0	0
string	0	1	1	1	1
0.5	0	0	0	0	0
empty	0	1	1	1	1
-1	1	0	0	0	0
2	1	1	1	1	1
string	1	0	0	0	0
0.5	1	1	1	1	1
empty	1	0	0	0	0
-1	2	0	1	1	0
2	2	0	0	0	1
string	2	0	0	0	0
0.5	2	1	0	0	1
empty	2	0	0	0	0
0	-1	1	1	1	1
1	-1	0	0	0	0
0	3	1	1	1	1
1	3	0	0	0	0
0	string	1	1	1	1
1	string	0	0	0	0
0	1.5	1	1	1	1
1	1.5	0	0	0	0
0	empty	1	1	1	1
1	empty	0	0	0	0
0	0	2	1	1	1
1	1	2	0	0	0
0	0	-1	1	1	1
1	1	-1	0	0	0
0	0	string	1	1	1
1	1	string	0	0	0
0	0	empty	1	1	1
1	1	empty	0	0	0
0	0	0.5	1	1	1
1	1	0.5	0	0	0
0	2	2	0	0	0
1	2	-1	0	0	0
1	2	string	0	0	1
1	2	empty	0	0	0

PGN:

Business:

- Event
  - a valid name of the Event where the game was played
- Site
  - a valid name of the site where the game was played
- Date
  - a valid date in the format YYYY.MM.DD
- Round
  - a valid Round of the event
- White
  - a valid name of the player that played with the white pieces
- Black
  - a valid name of the player that played with the black pieces
- Result
  - One of the acceptable results(1-0, 0-1, 1/2-1/2, *)
- PGN
  - Legal moves in order

Technical:

- Event, Site, Date, Round, White, Black, Result
  - string in double quotes
- PGN
  - Each turn has to have two legal moves
  - Each turn has to have a valid turn number, int > 0
  - At the end there has to be a result of the game

Event	Site	Date	Round	White	Black	Result	PGN
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	0	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	-1	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	NaN	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	empty	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	only one move in a turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	0-1	skip turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1/2-1/2	result doesnt match
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	more then 2 moves in one turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	illegal move
string (with quotes)	string (with quotes)	empty	1	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (no quotes)	1	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	InvalidDate	1	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	empty	1	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (no quotes)	1	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	empty	1	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (no quotes)	1	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	InvalidDate	1	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (no quotes)	1	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	InvalidDate	1	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	empty	1	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	InvalidDate	1	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	empty	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	string	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	0	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	-1	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	NaN	string (with quotes)	string (quotes)	*	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	-1	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	NaN	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	empty	string (with quotes)	string (quotes)	1-0	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	0	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	NaN	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	empty	string (with quotes)	string (quotes)	0-1	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	0	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	-1	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	empty	string (with quotes)	string (quotes)	1/2-1/2	valid
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	only one move in a turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	skip turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	result doesnt match
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	*	illegal move
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	skip turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	result doesnt match
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	1-0	more then 2 moves in one turn
string (with quotes)	string (with quotes)	ValidDate (quotes)	1	string (with quotes)	string (quotes)	0-1	only one move in a turn

TEST SCENARIO 1:

Atempting castling move in chess game

Description:
This integration test focuses on testing the integration and functionality of a chess application, specifically
the process starting from the game start until an attempt at castling. The purpose is to ensure that the chess pieces,
board state, and related rules are correctly implemented and integrated to support the castling move.

Test Steps:

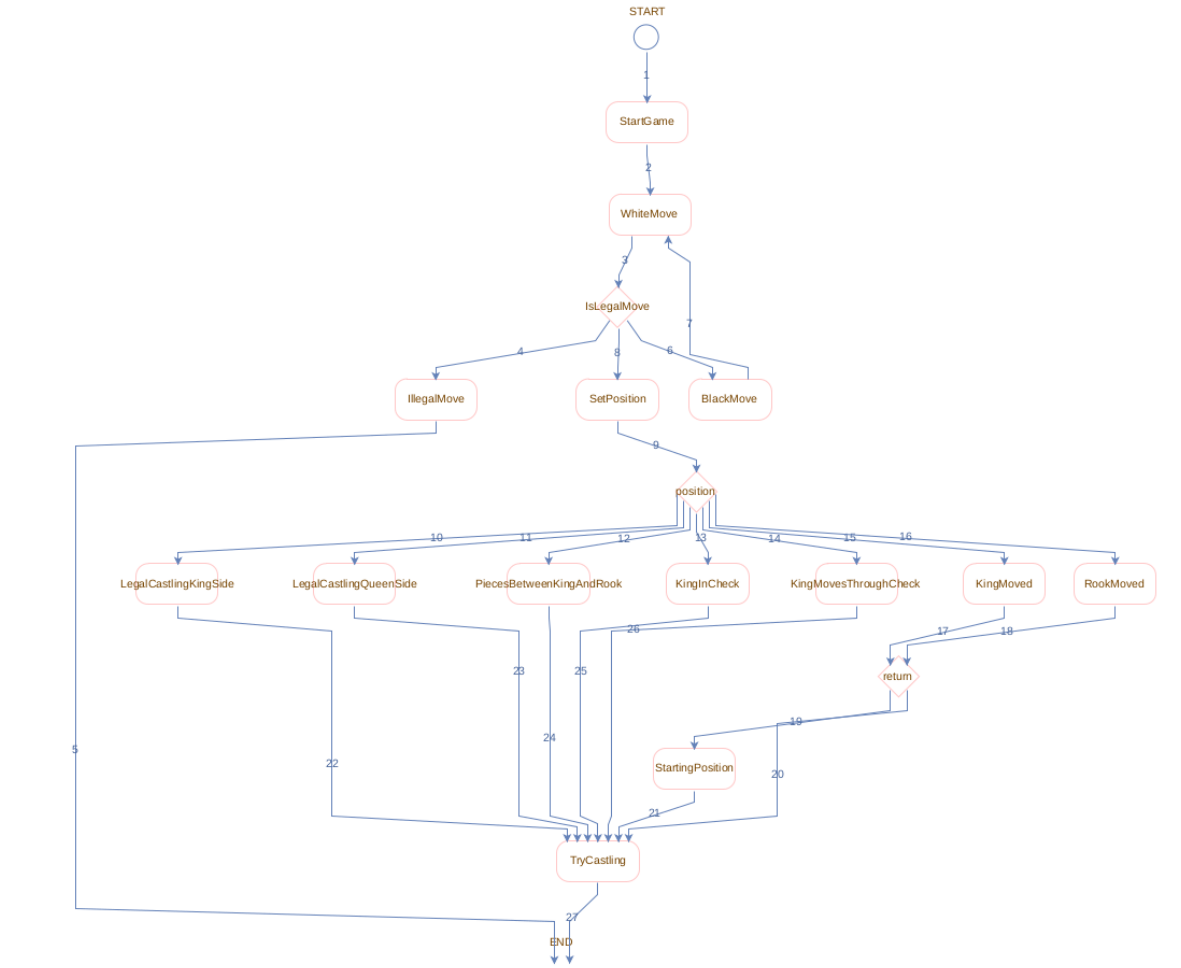
    Initialize the chess application and start a new game.
    Verify that the chessboard is set up correctly with the correct arrangement of pieces in their initial positions.
    Make the initial move for the white player, such as moving a pawn or a knight.
    Validate that the move is valid according to the rules of chess.
    Continue making moves for both players, to get to a point where the castling move is available.
    Perform the castling move.
    Validate that the castling move is executed correctly by updating the board state and moving the king and rook to their new positions.

Expected Results:

    The chessboard is set up correctly with the pieces in their initial positions.
    Moves made by players are valid and follow the rules of chess.
    The board state is updated accurately after each move.
    The castling move is only available under the appropriate conditions.
    Castling is executed correctly, resulting in the king and rook swapping positions.

Scenario: Illegal move.

```
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Attempt to perform an illegal move.
    Expected Result:
        The illegal move should fail.

Scenario: King has already moved.
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Move the king to a different position before attempting castling.
    Expected Result:
        The castling move should fail since the king has already moved.

Scenario: Rook has already moved.
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Move the rook to a different position before attempting castling.
    Expected Result:
        The castling move should fail since the rook has already moved.

Scenario: Pieces between the king and rook.
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Introduce one or more pieces (other than the king and rook) between
them.
    Expected Result:
        The castling move should fail since there are pieces obstructing
the path between the king and rook.

Scenario: King is in check.
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Place the king in a position where it is under threat (check).
    Expected Result:
        The castling move should fail since the king is in check.

Scenario: King moves through attacked squares.
    Steps:
        Perform moves to bring the game to a position suitable for
castling.
        Create a situation where the squares that the king will move
through during castling are under attack by opponent's pieces.
    Expected Result:
        The castling move should fail since the king would move through
attacked squares, which is not allowed.
```



Test situations 7, TDL= 2, ALG= PCT.

Sub-combinations of edges | Test situations

No.	Test sequence
1	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 4 - IllegalMove - 5 - END
2	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 6 - BlackMove - 7 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 10 - LegalCastlingKingSide - 22 - TryCastling - 27 - END
3	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 11 - LegalCastlingQueenSide - 23 - TryCastling - 27 - END
4	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 12 - PiecesBetweenKingAndRook - 24 - TryCastling - 27 - END
5	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 13 - KingInCheck - 25 - TryCastling - 27 - END
6	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 14 - KingMovesThroughCheck - 26 - TryCastling - 27 - END
7	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 15 - KingMoved - 17 - return - 19 - StartingPosition - 21 - TryCastling - 27 - END
8	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 16 - RookMoved - 18 - return - 19 - StartingPosition - 21 - TryCastling - 27 - END
9	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 15 - KingMoved - 17 - return - 20 - TryCastling - 27 - END
10	START - 1 - StartGame - 2 - WhiteMove - 3 - IsLegalMove - 8 - SetPosition - 9 - position - 16 - RookMoved - 18 - return - 20 - TryCastling - 27 - END

Situations: 10, Steps: 86, High: 0, Medium: 0, Low: 86, Unique nodes: 18, Unique edges: 27

TEST SCENARIO 2

Loading and Replay of Chess Game from a PGN File

Objective: The objective of this integration test is to ensure that the chess application correctly loads a PGN
(Portable Game Notation) file and successfully replays the chess game from that file.

Scenario: Play PGN, invalid header

- start application
- press play
- press load PGN game
- paste PGN with invalid header
- press Load
- should fail

Scenario: Replay PGN, wrong turn numbers
- start application
- open replay menu
- paste PGN with wrong turn numbers
- press Load
- should fail

Scenario: Replay PGN, invalid header
- start application
- open replay menu
- load a PGN file with invalid header
- should fail

Scenario: Play PGN, wrong turn numbers
- start application
- press play
- press load PGN game
- load a PGN file with wrong turn numbers
- press Load
- should fail

Scenario: Replay PGN, illegal sequence
- start application
- open replay menu
- paste PGN with illegal sequence
- press Load
- should fail

Scenario: Replay PGN, illegal move
- start application
- open replay menu
- paste PGN with illegal move
- press Load
- should fail

Scenario: Play PGN
- start application
- press play
- press load PGN game
- paste PGN
- press Load
- play a valid move
- should play a move

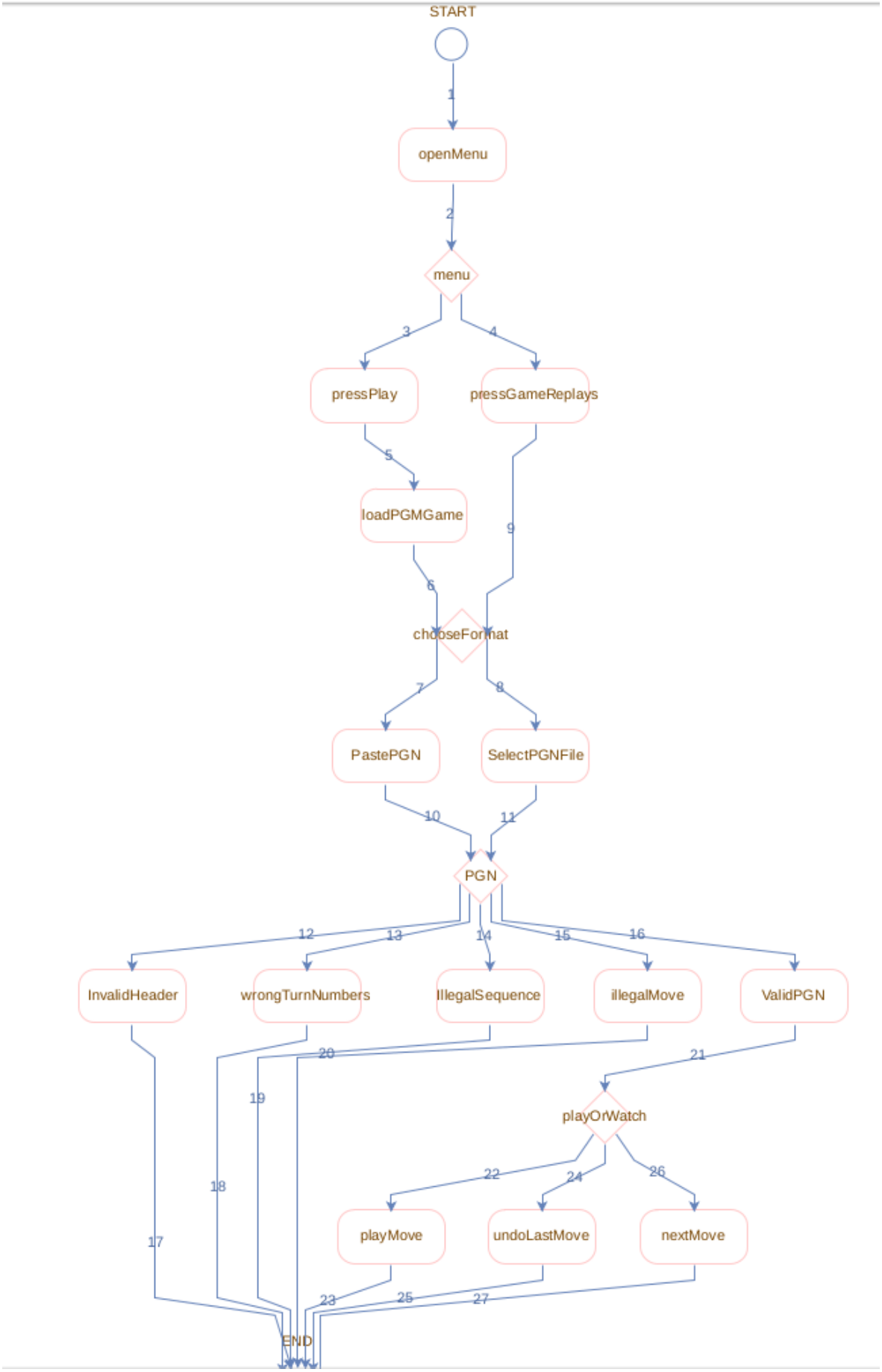
Scenario: Replay PGN, illegal sequence
- start application
- open replay menu
- load a PGN file with illegal sequence
- should fail

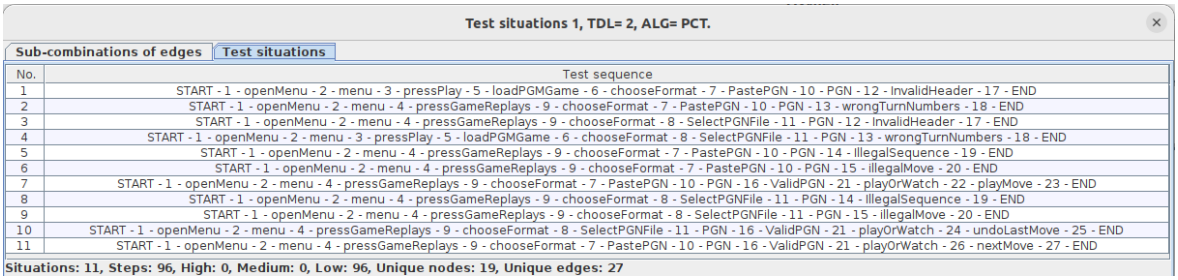
Scenario: Replay PGN, illegal move
- start application
- open replay menu
- load a PGN file with illegal move
- should fail

Scenario: Play PGN, undo last move
- start application
- press play
- press load PGN game
- load a PGN file
- press undo last move
- should undo last move

Scenario: Replay PGN, next move

- start application
- open replay menu
- paste PGN
- press next move
- should display next move


START
1
openMenu
2
menu
3
4
pressPlay
pressGameReplays
5
loadPGMGame
9
6
chooseFormat
7
8
PastePGN
SelectPGNFile
10
11
PGN
12
13
14
15
16
InvalidHeader
wrongTurnNumbers
IllegalSequence
illegalMove
ValidPGN
20
21
19
playOrWatch
22
24
26
18
17
playMove
undoLastMove
nextMove
23
25
27
END


Test situations 1, TDL= 2, ALG= PCT.
Sub-combinations of edges
Test situations
No.
Test sequence
1 START - 1 - openMenu - 2 - menu - 3 - pressPlay - 5 - loadPGMGame - 6 - chooseFormat - 7 - PastePGN - 10 - PGN - 12 - InvalidHeader - 17 - END
2 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 7 - PastePGN - 10 - PGN - 13 - wrongTurnNumbers - 18 - END
3 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 8 - SelectPGNFile - 11 - PGN - 12 - InvalidHeader - 17 - END
4 START - 1 - openMenu - 2 - menu - 3 - pressPlay - 5 - loadPGMGame - 6 - chooseFormat - 8 - SelectPGNFile - 11 - PGN - 13 - wrongTurnNumbers - 18 - END
5 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 7 - PastePGN - 10 - PGN - 14 - IllegalSequence - 19 - END
6 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 7 - PastePGN - 10 - PGN - 15 - illegalMove - 20 - END
7 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 7 - PastePGN - 10 - PGN - 16 - ValidPGN - 21 - playOrWatch - 22 - playMove - 23 - END
8 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 8 - SelectPGNFile - 11 - PGN - 14 - IllegalSequence - 19 - END
9 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 8 - SelectPGNFile - 11 - PGN - 15 - illegalMove - 20 - END
10 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 8 - SelectPGNFile - 11 - PGN - 16 - ValidPGN - 21 - playOrWatch - 24 - undoLastMove - 25 - END
11 START - 1 - openMenu - 2 - menu - 4 - pressGameReplays - 9 - chooseFormat - 7 - PastePGN - 10 - PGN - 16 - ValidPGN - 21 - playOrWatch - 26 - nextMove - 27 - END
Situations: 11, Steps: 96, High: 0, Medium: 0, Low: 96, Unique nodes: 19, Unique edges: 27