DESIGN DOCUMENT TEMPLATE

PROJECT GAME DESIGN

SECTION I: First Iteration Game Project Design [Till Phase III Submitted Upto 29th January, 2020]

PART I:

Sequence Diagrams - https://github.com/yashdhingra2198/FKApplyDesign/blob/Dev/sequenceDiagram.pdf
Class Diagrams - https://github.com/yashdhingra2198/FKApplyDesign/blob/Dev/classDiagram.pdf

PART II : Common Design/Choices and Conventions/Assumptions and Detailed Descriptions etc.

1. I thought of designing the v1.0 of tic tac toe with interface to implement the human play function and computer play function.
2. The tictactoe is limited to two players.
3. The users can play tic tac toe any number of times.
4. The board size will be taken by the user.
5. The decision to play human vs human or human vs computer will also be taken by user by providing the input.
6. The decision to play normal tic tac toe or hexagonal tic tac toe is also taken by user.
7. Human vs Human game
8. Human vs Computer game
9. Normal 3 * 3 or 4 * 4 tic tac toe game

10. Enhanced tic tac toe game (board size to be multiple of 3 like 9 * 9 and so on)
11. Players can go multiple level in enhanced tic tac toe game
12. Super titactoe game with regular hexagonal board
13. Leaderboard scores

PART III : Feature Specific Design/Choices and Conventions/Assumptions

GameDesign v2.0 - Requirement I

<COMPLETE> - 1. Tic-Tac-Toe consists of 3x3 Square Cells
Designed n*n size board rather than restricting it to 3*3

<COMPLETE> - 2. Game Between Two Humans
Implemented an interface to write the play method for human

<COMPLETE> - 3. Game Between Human and Machine
Implemented an interface to write the play method for computer

<COMPLETE> - 4. Winning Criteria - 3 Cells in Row/Column/Diagonal are in Same State.
Checkwin function implements the winning criteria and checks the winner after every move

<COMPLETE> - 5. Announce Winning Player

GameDesign v2.0 - Requirement II

<COMPLETE> - 6. Enhanced Tic-Tac-Toe Game Consist of 9x9 Squares...
Implemented the 9*9 tic tac toe using the same code

<COMPLETE> - 7. Enhanced Tic-Tac-Toe will continue to expand in depth levels...
3D board is taken to implement the levels in the game and stores state of every move in every level

<COMPLETE> - 8. Extend Game to 4x4 Board
Implemented with the same design as 3*3

<PARTIALLY COMPLETE> - 9. Human Player is Biased...

<PARTIALLY COMPLETE> - 10. Storing and Retrieving Game State
Winner is check after every move and if winner is still null the game is played else winner is declared.

<COMPLETE> - 11. Store Players Game Statistics: Leaderboard
Leaderboard is shown when either the match is won or draws.

GameDesign v3.0 - Requirement III

<COMPLETE> - 12. Super Tic-Tac-Toe Game Extends Enhanced Tic-Tac-Toe Game...
Manager interface is implemented to handle the normal tic tac toe and super tic tac toe(HEXAGONAL)

<COMPLETE> - 13. Design Winning and Losing Criterias On All Edges...
Checkwin function for super tic tac toe checks the condition by

further calling the check functions on row and the tow diagonals.

<NOT COMPLETE> - 14. Incorporate Irregular shaped Hexagonal Boards

GameDesign v4.0 - Requirement IV

<NOT COMPLETE> - 15. Incorporate Biased Game Board

<NOT COMPLETE> - 16.Incorporate Connect Four Game In Design

<NOT COMPLETE> - 17. Discover Newer Abstract Types

<NOT COMPLETE> - 18. Refactor and Reuse Code In Both Games

SECTION II: Second Iteration[Refactoring/Redesign] Game Project Design

[Till Phase III Submitted Upto 03rd February, 2020]

PART I : Common Design/Choices, Conventions and Assumptions

Sequence Diagrams - https://github.com/yashdhingra2198/FKApplyDesign/blob/Dev/sequenceDiagram.pdf
Class Diagrams - https://github.com/yashdhingra2198/FKApplyDesign/blob/Dev/classDiagram.pdf

PART II : Common Design/Choices and Conventions/Assumptions and Detailed Descriptions etc.

1. Refactored the code design.
2. Removed nested if-else.
3. Some objects which were created in every move are handled properly.
4. Implemented the row check and diagonal check function for hexagonal tic tac toe and called these function from the checkwin function.

PART III: Feature Specific Design/Choices, Conventions and Assumptions

GameDesign v2.0 - Requirement I

<COMPLETE> - 1. Tic-Tac-Toe consists of 3x3 Square Cells

Designed n*n size board rather than restricting it to 3*3

<COMPLETE> - 2. Game Between Two Humans

Implemented an interface to write the play method for human

<COMPLETE> - 3. Game Between Human and Machine

Implemented an interface to write the play method for computer

<COMPLETE> - 4. Winning Criteria - 3 Cells in Row/Column/Diagonal are in Same State.

Checkwin function implements the winning criteria and checks the winner after every move

<COMPLETE> - 5. Announce Winning Player

GameDesign v2.0 - Requirement II

<COMPLETE> - 6. Enhanced Tic-Tac-Toe Game Consist of 9x9 Squares...

Implemented the 9*9 tic tac toe using the same code

<COMPLETE> - 7. Enhanced Tic-Tac-Toe will continue to expand in depth levels...

3D board is taken to implement the levels in the game and stores state of every move in every level

<COMPLETE> - 8. Extend Game to 4x4 Board

Implemented with the same design as 3*3

<PARTIALLY COMPLETE> - 9. Human Player is Biased...

<PARTIALLY COMPLETE> - 10. Storing and Retrieving Game State

Winner is check after every move and if winner is still null the game is played else winner is declared.

<COMPLETE> - 11. Store Players Game Statistics: Leaderboard

Leaderboard is shown when either the match is won or draws.

GameDesign v3.0 - Requirement III

<COMPLETE> - 12. Super Tic-Tac-Toe Game Extends Enhanced Tic-Tac-Toe Game...

Manager interface is implemented to handle the normal tic tac toe and super tic tac toe(HEXAGONAL)

<COMPLETE> - 13. Design Winning and Losing Criterias On All Edges...

Checkwin function for super tic tac toe checks the condition by further calling the check functions on row and the tow diagonals.

<NOT COMPLETE> - 14. Incorporate Irregular shaped Hexagonal Boards

GameDesign v4.0 - Requirement IV

<NOT COMPLETE> - 15. Incorporate Biased Game Board

<NOT COMPLETE> - 16.Incorporate Connect Four Game In Design

<NOT COMPLETE> - 17. Discover Newer Abstract Types

<NOT COMPLETE> - 18. Refactor and Reuse Code In Both Games

SECTION III: GameDesign Project Feature and Test Cases Implementation Status and

Description

GameDesign v1.0 - Requirement I

Completed - 1. Tic-Tac-Toe consists of 3x3 Square Cells

TestCases:

Test1 : Test Generation of 3x3 Board

Class Name- HumanTest , Function Names- testAdd()

Test2 : Test Initialisation of 3x3 Board

Class Name- HumanTest , Function Names- testAdd()

Test3 : Test Cell Generation

Class Name- HumanTest , Function Names- testAdd()

Completed - 2. Game Between Two Humans

TestCases:

Test1 : Test Generation of 3x3 Board

Class Name- HumanTest , Function Names- testAdd()

Test2 : Test Initialisation of 3x3 Board

Class Name- HumanTest , Function Names- testAdd()

Test3 : Test Cell Generation

Class Name- HumanTest , Function Names- testAdd()

Not Completed the test cases - 3. Game Between Human and Machine

Not Completed the test cases - 4. Winning Criteria - 3 Cells in Row/Column/Diagonal are in the Same State.

Not Completed the test cases - 5. Announce Winning Player

Not Completed the test cases for GameDesign v2.0 - Requirement II

Not Completed the test cases for GameDesign v3.0 - Requirement III

Not Completed the test cases for GameDesign v4.0 - Requirement IV

SECTION III: How to Run/Test Your Code?

1. Clone the project from https://github.com/yashdhingra2198/FKApplyDesign/tree/Dev
2. The main file is tictactoe.java
3. Type the following command in terminal
   To compile- javac tictactoe.java -d classFiles
   To run - java -cp classFiles/ tictactoe

You can run the HumanTest.java to rest the 3*3 board functionalities like initialiseBoard , checkWin function and printBoard function

Are you providing all Input/Output files to run Test Code using Test.java?
No, input/output file is required for HumanTest.java