Design Document TicTacToe V1

By Karunam Goyal #138935

FKApplyDesign

How to Run

To Run

Clone https://github.com/karunamgoyal/FKApplyDesign.git

git checkout dev

Then Run Shell Script It Will Automatically Compile All the Files

./Compile.sh

Then To Run the Game Type

./Run

To Run Test First Compile Test with

./Compile.sh

then

./RunTest.sh

In your terminal

Enter Coordinates of the box and top left corner being 0 0 (Basic Coordinate Convention for Matrix)

Directory Structure of Java Files

```
+FKApplyDesign
  -+src

   -+Players
    - Player.java
    - Machine.java
    - Human.java
   -+State
    - LeaderBoard.java
    - LeaderBoardWhole.java
    - Hex.java
    - StateInterface.java
    - State.java
    - StateManager.java
    - StateUpdater.java
    - Point.java
   - PlayGround.java
   - Run.java
  - Compile.sh
  - Run.sh
```

Directory Structure of Class Files

```
+FKApplyDesign
  -+src
   -+Players
    - Player.class
    - Machine.class
    - Human.class
   -+State
    - LeaderBoard.java
    - LeaderBoardWhole.java
    - Hex.java
    - StateInterface.java
    - State.class
    - StateManager.class
    - StateUpdater.class
    - Point.class
   -+Playground
    - PlayGround.class
   - Run.class
  - Run.sh
```

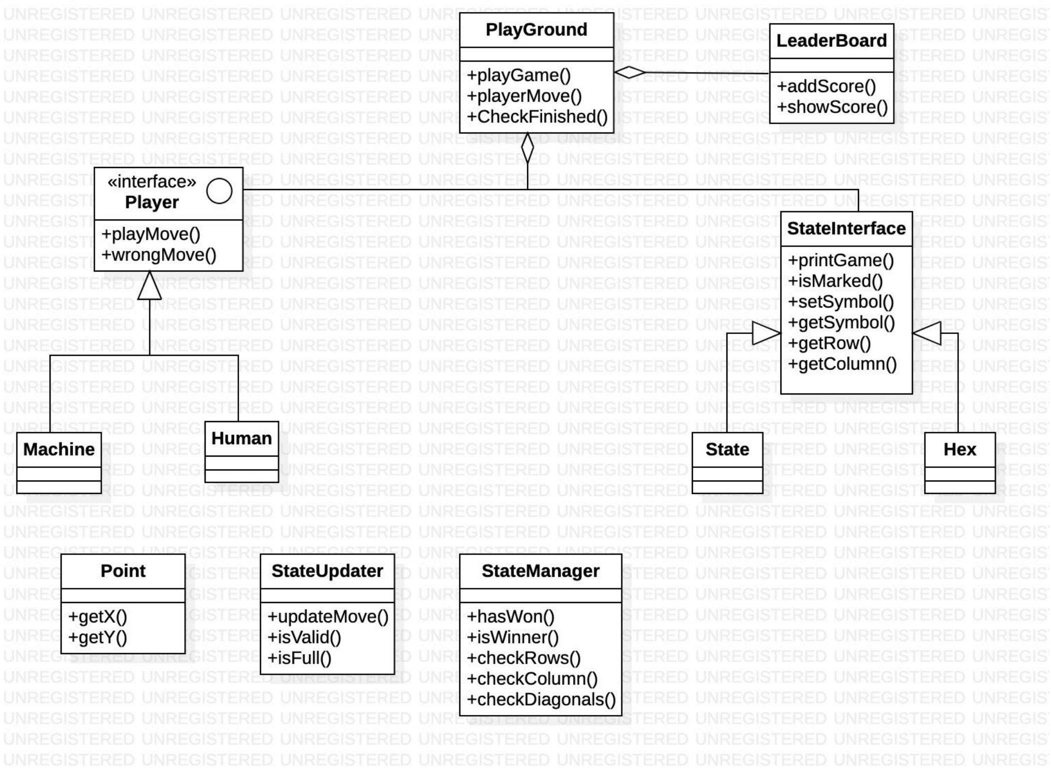
Directory Structure of TestFiles

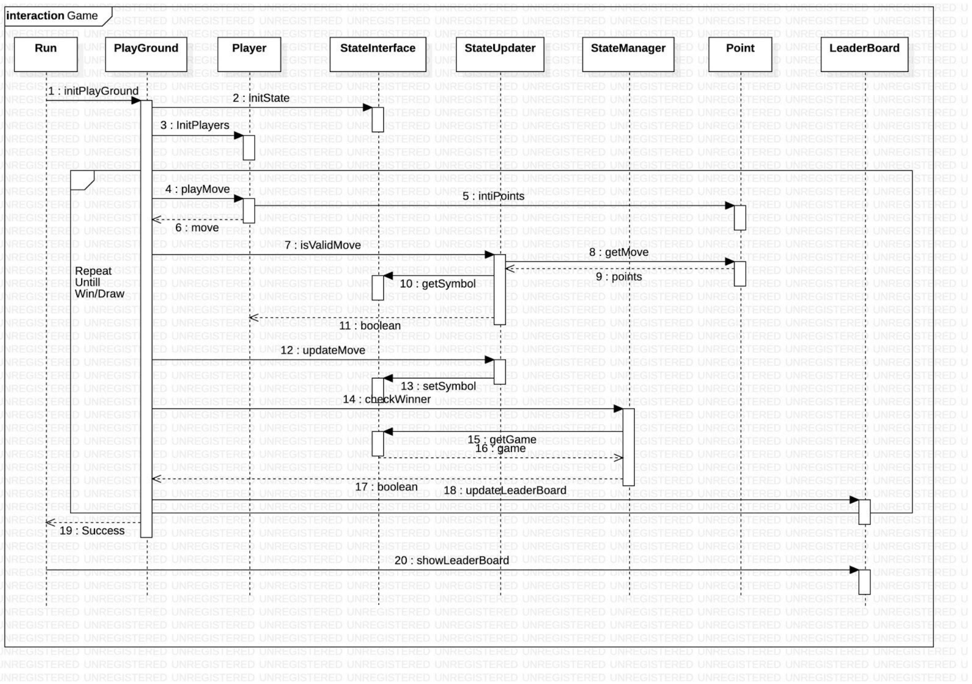
```
+FKApplyDesign
  -+src
   - MachineTest.java
   - HexTest.java
   - StateTest.java
  - RunTest.sh
  - CompileTest.sh
  -junit.jar
  -hamcrest.jar
```

Maintainability

CodeBase is Managed over multiple directories so it is easy for it to manage and maintain the Code for extension and later versions.

Design Plan





```
Instance of a Game Being Played at a Moment
|''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''|
| |''''''''| |'''''''''|         |''''''''''''''|  |'''''''''''| |
| |Player1 | | Player2 |         |StateInterface|  |LeaderBoard| |
| |        | |         |         |              |  |           | |
| '''''''''' '''''''''''         ''''''''''''''''  ''''''''''''' |
|                                                                |
|                |''''''''''''''| |'''''''''''''|                |
|                | State Updater| | StateManager|                |
|                |              | |             |                |
|                '''''''''''''''' '''''''''''''''                |
|                           PlayGround                           |
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Playground Class Which Act As A Box Where Game is Played

LeaderBoard Maintains LeaderBoard

Player is an Interface act as a Base for Human and Machine Player

Human and Machine Class Plays their Moves and they have a symbol they play

StateInterface Act as an Abstract class for TicTacToe and Hex

StateUpdater as the name Suggests Contains a State Variable and Updates the state and checks its value

StateManager manages the State and Checks if Someone Has Won and returns accordingly

PlayGround Contains the Objects of Players, State, Manager, Updater and the game is played and it act as a box

Run Runs The PlayGround

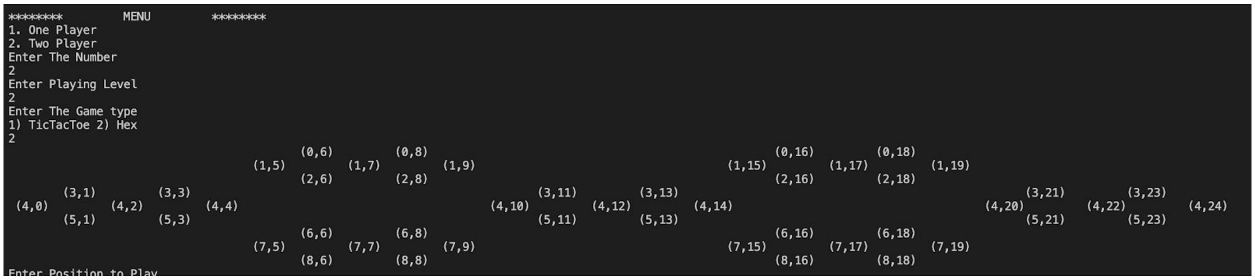
Functionality

- Multiple Players
- M X M Row For both TicTacToe and Hex
- Players can go to multiple Levels
- Scoring based on level
- bias in the move Revert move

Test

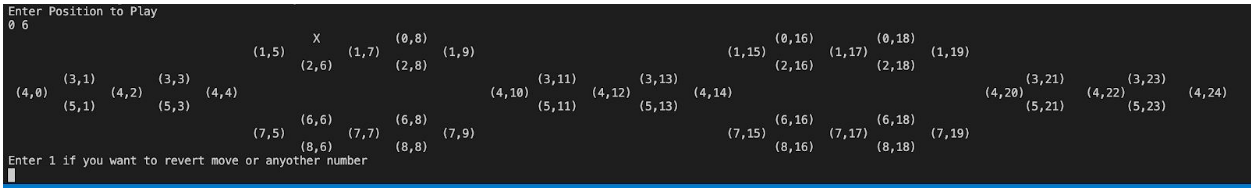
```
Added Test using Junit and testing most critical classes and also including corner cases
```

Sample Inputs



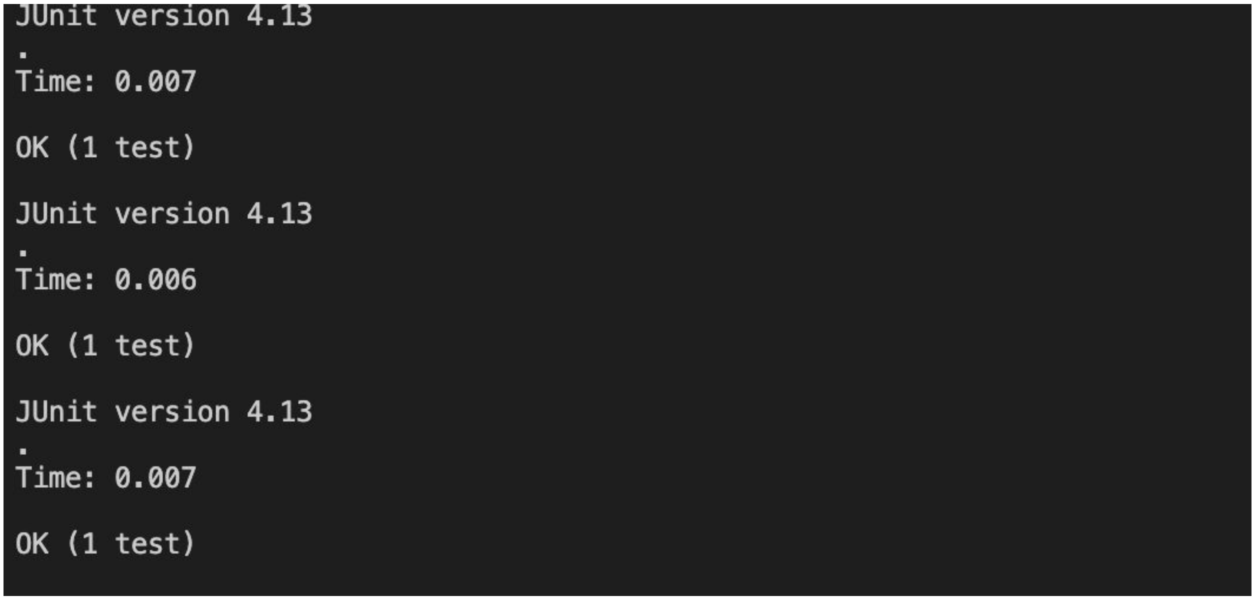
```
********        MENU        ********
1. One Player
2. Two Player
Enter The Number
2
Enter Playing Level
2
Enter The Game type
1) TicTacToe 2) Hex
2
                                           (0,6)         (0,8)                                                   (0,16)         (0,18)
                                    (1,5)         (1,7)         (1,9)                                     (1,15)         (1,17)         (1,19)
                                           (2,6)         (2,8)                                                   (2,16)         (2,18)
        (3,1)         (3,3)                                                   (3,11)         (3,13)                                                   (3,21)         (3,23)
 (4,0)         (4,2)         (4,4)                                     (4,10)         (4,12)         (4,14)                                     (4,20)         (4,22)         (4,24)
        (5,1)         (5,3)                                                   (5,11)         (5,13)                                                   (5,21)         (5,23)
                                           (6,6)         (6,8)                                                   (6,16)         (6,18)
                                    (7,5)         (7,7)         (7,9)                                     (7,15)         (7,17)         (7,19)
                                           (8,6)         (8,8)                                                   (8,16)         (8,18)

Enter Position to Play
```



```
Enter Position to Play
0 6
                                             X           (0,8)                                                   (0,16)         (0,18)
                                    (1,5)         (1,7)         (1,9)                                     (1,15)         (1,17)         (1,19)
                                           (2,6)         (2,8)                                                   (2,16)         (2,18)
        (3,1)         (3,3)                                                   (3,11)         (3,13)                                                   (3,21)         (3,23)
 (4,0)         (4,2)         (4,4)                                     (4,10)         (4,12)         (4,14)                                     (4,20)         (4,22)         (4,24)
        (5,1)         (5,3)                                                   (5,11)         (5,13)                                                   (5,21)         (5,23)
                                           (6,6)         (6,8)                                                   (6,16)         (6,18)
                                    (7,5)         (7,7)         (7,9)                                     (7,15)         (7,17)         (7,19)
                                           (8,6)         (8,8)                                                   (8,16)         (8,18)
Enter 1 if you want to revert move or anyother number
```

Test Results are OK Passed



```
JUnit version 4.13
.
Time: 0.007

OK (1 test)

JUnit version 4.13
.
Time: 0.006

OK (1 test)

JUnit version 4.13
.
Time: 0.007

OK (1 test)
```

Why I Think This Design is better

This Design is Good because I've Tried To Do Loose Coupling and use Composition rather than inheriting the things.

This can be further extended to play on multiple systems after adding locking mechanisms or similar Techniques

> The System After refactoring is very loosely coupled
> Updated After refactoring wherever necessary