Package edu.cmu.cs780.hw3

Class Game

java.lang.Object
    edu.cmu.cs780.hw3.Game

```
public class Game
extends Object
```

Homework 3

Author:

Zihao Yang (zihaoy2@andrew.cmu.edu), Yun Lee (yunl3@andrew.cmu.edu)

Constructor Summary

Constructors

Constructor	Description
`Game()`	Initializes the game with an empty board, sets the checkers count to zero, and randomly decides the starting player.
`Game(int[][] newBoard, int checkersCount, int currentPlayer)`	Constructor to create a Game instance using a provided game board, the current number of checkers, and the starting player.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
boolean	`canGameContinue()`	Determines if the game can continue or if it has reached an end condition.
boolean	`hasWinner()`	Determines if there is a winner based on the current game board state.
boolean	`hasWinner(int[][] gameBoard)`	Determines if there is a winner on a given game board.
void	`placeChecker(int col)`	Places a checker in the specified column.
String	`prettyPrintBoard()`	Transform the board into a string to be

printed in the terminal.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Game

```
public Game()
```

Initializes the game with an empty board, sets the checkers count to zero, and randomly decides the starting player.

Game

```
public Game(int[][] newBoard,
            int checkersCount,
            int currentPlayer)
```

Constructor to create a Game instance using a provided game board, the current number of checkers, and the starting player.

Parameters:

`newBoard` - Input game board represented as a 2D array.

`checkersCount` - Current number of checkers on the board.

`currentPlayer` - The ID of the player set to play next.

Method Details

canGameContinue

```
public boolean canGameContinue()
```

Determines if the game can continue or if it has reached an end condition. This method also prints out the game status to the console.

Returns:

Returns true if the game can continue, false if the game has reached a draw or a win condition.

prettyPrintBoard

```
public String prettyPrintBoard()
```

Transform the board into a string to be printed in the terminal.

Returns:

Returns a String representation of pretty printed board.

hasWinner

```
public boolean hasWinner()
```

Determines if there is a winner based on the current game board state.

Returns:

Returns true if the current player wins, false otherwise.

hasWinner

```
public boolean hasWinner(int[][] gameBoard)
```

Determines if there is a winner on a given game board.

Parameters:

`gameBoard` - Input Game board represented as a 2D array.

Returns:

Returns true if there is a winner, false otherwise.

placeChecker

```
public void placeChecker(int col)
```

Places a checker in the specified column. If the chosen column is valid (not full), the current player's checker is added to the game board and the role is switched to the other player. If the chosen column is already full, an appropriate message is displayed.

Parameters:

`col` - The index-based column number where the current player wants to place their checker.