Package **Class** **Tree** **Deprecated** **Index** **Help**

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Interface IGameLogic

All Known Implementing Classes:

GameLogic

```
public interface IGameLogic
```

Method Summary

`int`	`deciteNextMove()` Calculates the next move This is where you should implement/call your heuristic evaluation functions etc.
`int`	`gameFinished()` Checks if any of the two players have 4-in-a-row.
`void`	`initializeGame(int columns, int rows, int playerID)` Creates a new empty game board of the specified dimensions and indicates the ID of the player.
`void`	`insertCoin(int column, int playerID)` Notifies that a token/coin is put in the specified column of the game board.

Method Detail

initializeGame

```
void initializeGame(int columns,
                    int rows,
                    int playerID)
```

Creates a new empty game board of the specified dimensions and indicates the ID of the player. This method will be called from the main function.

Parameters:

`columns` - The number of columns in the game board
`rows` - The number of rows in the game board

`playerID` - 1 = blue (player1), 2 = red (player2)

---

insertCoin

```
void insertCoin(int column,
                int playerID)
```

Notifies that a token/coin is put in the specified column of the game board.

Parameters:

`column` - The column where the coin is inserted.
`playerID` - The ID of the current player.

---

decideNextMove

```
int decideNextMove()
```

Calculates the next move This is where you should implement/call your heuristic evaluation functions etc.

---

gameFinished

```
int gameFinished()
```

Checks if any of the two players have 4-in-a-row.

Returns:

1 if blue won, 2 if red won and 0 if it is a tie.
