JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/ClearCellGame.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/model/BoardCell.html)
* [Next Class](http://docs.google.com/model/Game.html)
* [Frames](http://docs.google.com/index.html?model/ClearCellGame.html)
* [No Frames](http://docs.google.com/ClearCellGame.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)

model

## Class ClearCellGame

* java.lang.Object
  + [model.Game](http://docs.google.com/model/Game.html)
    - model.ClearCellGame
* public class ClearCellGame  
  extends [Game](http://docs.google.com/model/Game.html)  
  This class extends GameModel and implements the logic of the clear cell game. We define an empty cell as BoardCell.EMPTY. An empty row is defined as one where every cell corresponds to BoardCell.EMPTY.Author: Dept of Computer Science, UMCP

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**ClearCellGame**](http://docs.google.com/model/ClearCellGame.html#ClearCellGame(int,%20int,%20java.util.Random,%20int))(int maxRows, int maxCols, java.util.Random random, int strategy) Defines a board with empty cells. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + int | * + [**getScore**](http://docs.google.com/model/ClearCellGame.html#getScore())() |
| * + boolean | * + [**isGameOver**](http://docs.google.com/model/ClearCellGame.html#isGameOver())() The game is over when the last board row (row with index board.length -1) is different from empty row. |
| * + void | * + [**nextAnimationStep**](http://docs.google.com/model/ClearCellGame.html#nextAnimationStep())() This method will attempt to insert a row of random BoardCell objects if the last board row (row with index board.length -1) corresponds to the empty row; otherwise no operation will take place. |
| * + void | * + [**processCell**](http://docs.google.com/model/ClearCellGame.html#processCell(int,%20int))(int rowIndex, int colIndex) This method will turn to BoardCell.EMPTY the cell selected and any adjacent surrounding cells in the vertical, horizontal, and diagonal directions that have the same color. |

### Methods inherited from class model.[**Game**](http://docs.google.com/model/Game.html)[getBoardCell](http://docs.google.com/model/Game.html#getBoardCell(int,%20int)), [getMaxCols](http://docs.google.com/model/Game.html#getMaxCols()), [getMaxRows](http://docs.google.com/model/Game.html#getMaxRows()), [setBoardCell](http://docs.google.com/model/Game.html#setBoardCell(int,%20int,%20model.BoardCell)), [setBoardWithColor](http://docs.google.com/model/Game.html#setBoardWithColor(model.BoardCell)), [setColWithColor](http://docs.google.com/model/Game.html#setColWithColor(int,%20model.BoardCell)), [setRowWithColor](http://docs.google.com/model/Game.html#setRowWithColor(int,%20model.BoardCell))

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### ClearCellGame public ClearCellGame(int maxRows, int maxCols, java.util.Random random, int strategy) Defines a board with empty cells. It relies on the super class constructor to define the board. The random parameter is used for the generation of random cells. The strategy parameter defines which clearing cell strategy to use (for this project it will be 1). For fun, you can add your own strategy by using a value different that one.Parameters:maxRows - maxCols - random - strategy -

### Method Detail

#### isGameOver public boolean isGameOver() The game is over when the last board row (row with index board.length -1) is different from empty row.**Specified by:** [isGameOver](http://docs.google.com/model/Game.html#isGameOver()) in class [Game](http://docs.google.com/model/Game.html)

#### getScore public int getScore()**Specified by:** [getScore](http://docs.google.com/model/Game.html#getScore()) in class [Game](http://docs.google.com/model/Game.html)

#### nextAnimationStep public void nextAnimationStep() This method will attempt to insert a row of random BoardCell objects if the last board row (row with index board.length -1) corresponds to the empty row; otherwise no operation will take place.**Specified by:** [nextAnimationStep](http://docs.google.com/model/Game.html#nextAnimationStep()) in class [Game](http://docs.google.com/model/Game.html)

* + - processCell  
      public void processCell(int rowIndex,  
       int colIndex)  
      This method will turn to BoardCell.EMPTY the cell selected and any adjacent surrounding cells in the vertical, horizontal, and diagonal directions that have the same color. The clearing of adjacent cells will continue as long as cells have a color that corresponds to the selected cell. Notice that the clearing process does not clear every single cell that surrounds a cell selected (only those found in the vertical, horizontal or diagonal directions). IMPORTANT: Clearing a cell adds one point to the game's score.  
        
        
        
      If after processing cells, any rows in the board are empty,those rows will collapse, moving non-empty rows upward. For example, if we have the following board (an \* represents an empty cell):  
        
        
        
      RRR  
        
      GGG  
        
      YYY  
        
      \* \* \*  
        
        
        
      then processing each cell of the second row will generate the following board  
        
        
        
      RRR  
        
      YYY  
        
      \* \* \*  
        
      \* \* \*  
        
        
        
      IMPORTANT: If the game has ended no action will take place.**Specified by:** [processCell](http://docs.google.com/model/Game.html#processCell(int,%20int)) in class [Game](http://docs.google.com/model/Game.html)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/ClearCellGame.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/model/BoardCell.html)
* [Next Class](http://docs.google.com/model/Game.html)
* [Frames](http://docs.google.com/index.html?model/ClearCellGame.html)
* [No Frames](http://docs.google.com/ClearCellGame.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)