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/Game.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/ClearCellGame.html)
* Next Class
* [Frames](http://docs.google.com/index.html?model/Game.html)
* [No Frames](http://docs.google.com/Game.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#4d34og8)

model

## Class Game

* java.lang.Object
  + model.Game
* Direct Known Subclasses: [ClearCellGame](http://docs.google.com/model/ClearCellGame.html)  
    
  public abstract class Game  
  extends java.lang.Object  
  This class represents the logic of a game where a board is updated on each step of the game animation. The board can also be updated by selecting a board cell.Author: Dept of Computer Science, UMCP

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**Game**](http://docs.google.com/model/Game.html#Game(int,%20int))(int maxRows, int maxCols) Defines a board with BoardCell.EMPTY cells. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [BoardCell](http://docs.google.com/model/BoardCell.html) | * + [**getBoardCell**](http://docs.google.com/model/Game.html#getBoardCell(int,%20int))(int rowIndex, int colIndex) |
| * + int | * + [**getMaxCols**](http://docs.google.com/model/Game.html#getMaxCols())() |
| * + int | * + [**getMaxRows**](http://docs.google.com/model/Game.html#getMaxRows())() |
| * + abstract int | * + [**getScore**](http://docs.google.com/model/Game.html#getScore())() |
| * + abstract boolean | * + [**isGameOver**](http://docs.google.com/model/Game.html#isGameOver())() |
| * + abstract void | * + [**nextAnimationStep**](http://docs.google.com/model/Game.html#nextAnimationStep())() Advances the animation one step. |
| * + abstract void | * + [**processCell**](http://docs.google.com/model/Game.html#processCell(int,%20int))(int rowIndex, int colIndex) Adjust the board state according to the current board state and the selected cell. |
| * + void | * + [**setBoardCell**](http://docs.google.com/model/Game.html#setBoardCell(int,%20int,%20model.BoardCell))(int rowIndex, int colIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) boardCell) |
| * + void | * + [**setBoardWithColor**](http://docs.google.com/model/Game.html#setBoardWithColor(model.BoardCell))([BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes the board with the specified color. |
| * + void | * + [**setColWithColor**](http://docs.google.com/model/Game.html#setColWithColor(int,%20model.BoardCell))(int colIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes column with the specified color. |
| * + void | * + [**setRowWithColor**](http://docs.google.com/model/Game.html#setRowWithColor(int,%20model.BoardCell))(int rowIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes row with the specified color. |

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

### Constructor Detail

#### Game public Game(int maxRows, int maxCols) Defines a board with BoardCell.EMPTY cells.Parameters:maxRows - maxCols -

### Method Detail

#### getMaxRows public int getMaxRows()

#### getMaxCols public int getMaxCols()

#### setBoardCell public void setBoardCell(int rowIndex, int colIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) boardCell)

#### getBoardCell public [BoardCell](http://docs.google.com/model/BoardCell.html) getBoardCell(int rowIndex, int colIndex)

#### setRowWithColor public void setRowWithColor(int rowIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes row with the specified color.Parameters:rowIndex - cell -

#### setColWithColor public void setColWithColor(int colIndex, [BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes column with the specified color.Parameters:colIndex - cell -

#### setBoardWithColor public void setBoardWithColor([BoardCell](http://docs.google.com/model/BoardCell.html) cell) Initializes the board with the specified color.Parameters:cell -

#### isGameOver public abstract boolean isGameOver()

#### getScore public abstract int getScore()

#### nextAnimationStep public abstract void nextAnimationStep() Advances the animation one step.

#### processCell public abstract void processCell(int rowIndex, int colIndex) Adjust the board state according to the current board state and the selected cell.Parameters:rowIndex - colIndex -

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Game.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/ClearCellGame.html)
* Next Class
* [Frames](http://docs.google.com/index.html?model/Game.html)
* [No Frames](http://docs.google.com/Game.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#4d34og8)