|  |
| --- |
| GameGrid |
| - [**bgColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#bgColor) **:** java.awt.Color  The color of the background.  **-** [**gridColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#gridColor)**:** java.awt.Color  Color of the grid.  **-** [**bgImagePath**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#bgImagePath)**:** java.lang.String  Path of the background image.  **-** [**BLACK**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#BLACK) **/** [**LIGHTGRAY**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#LIGHTGRAY) **/ WHITE / … :** static java.awt.Color  Short for Color.Black/Lightgray,white, red, …  **-** [**cellSize**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#cellSize)**: int**  Size of cells in pixels (square).  - [**nbHorzCells**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#nbHorzCells)**: int**  Number of horizonal cells.  **-** [**nbVertCells**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#nbVertCells)**: int**  Number of vertical cells.  **-** [**monitor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#monitor)**:** static java.lang.Object  Object used for synchronizing access of methods accessing the sceen list. |
| **Constructors:**  [**GameGrid**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#GameGrid(int, int, int))(int nbHorzCells, int nbVertCells, int cellSize)  Constructs a game window including a playground with a navigation bar and no visible grid and no background image.  [**GameGrid**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#GameGrid(int, int, int, java.awt.Color, java.lang.String))(int nbHorzCells, int nbVertCells, int cellSize, java.awt.Color gridColor, java.lang.String bgImagePath)  Constructs a game window including a playground with a navigation bar, possibly a visible grid and possibly a background image.  [**GameGrid**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#GameGrid(int, int, int, java.lang.String))(int nbHorzCells, int nbVertCells, int cellSize, java.lang.String bgImagePath)  Constructs a game window including a playground with a navigation bar and a possibly a background image, but no visible grid. |
| **Methods:**  [**act**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#act())(): void  Empty method called in every simulation cycle after all actor act() calls.  [**actAll**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#actAll())(): void  Invokes all actor's act() methods in the order of the scene and draws the new game situation.  [**activate**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#activate())(): void  Brings the game grid window to the front and request the focus.  [**addActorNoRefresh**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#addActorNoRefresh(ch.aplu.jgamegrid.Actor, ch.aplu.jgamegrid.Location))([**Actor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/Actor.html) actor, [**Location**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/Location.html) location): void  Same as addActor(Actor actor, Location location), but the game grid window is not refreshed automatically.  [**addExitListener**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#addExitListener(ch.aplu.jgamegrid.GGExitListener))(**[GGExitListener](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GGExitListener.html" \o "interface in ch.aplu.jgamegrid)** listener): void  Registers the given GGExitListener to get an event when the title bar close button is hit.  [**addKeyListener**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#addKeyListener(ch.aplu.jgamegrid.GGKeyListener))(**[GGKeyListener](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GGKeyListener.html" \o "interface in ch.aplu.jgamegrid)** listener):void  Adds a GGKeyListener to get events when a key is pressed.  [**addStatusBar**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#addStatusBar(int))(int height)  Adds a status window attached at the bottom of the game grid window (below the navigation bar, if available).  [**dispose**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#dispose())(): void  Stops the game thread, hides the game window, disposes the frame and calls Monitor.wakeUp().  [**doPause**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#doPause())(): void  Pauses the simulation cycling.  [**doRun**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#doRun())(): void  Starts the simulation cycling.  [**getActors**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#getActors())():java.util.ArrayList<[**Actor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/Actor.html)>  Returns all actors that are part of the game grid.  [**getBgColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#getBgColor())():java.awt.Color  Returns the color of the background.  [**getOccupiedLocations**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#getOccupiedLocations())():java.util.ArrayList<[**Location**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/Location.html)>  Returns a list of all locations occupied by actors.  [**getRandomEmptyLocation**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#getRandomEmptyLocation())()  Returns an empty random location within the game grid.  [**hide**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#hide())(): void  Hides the game grid, but does not destroy it.  [**isAtBorder**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#isAtBorder(ch.aplu.jgamegrid.Location))([**Location**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/Location.html) location): boolean  Returns true, if the given cell location is at the grid border.  [**refresh**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#refresh())(): void  Refreshs the current game situation (repaint background, tiles, actors).  [**setBgColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setBgColor(java.awt.Color))(java.awt.Color color): void  Sets the color of the background.  [**setBgColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setBgColor(int, int, int))(int r, int g, int b): void  Sets the color of the background.  [**setBgImagePath**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setBgImagePath(java.lang.String))(java.lang.String bgImagePath): void  Sets the path to the background image file.  [**setGridColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setGridColor(java.awt.Color))(java.awt.Color color): void  Sets the color of the grid and reconstructs the background.  [**setTitle**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setTitle(java.lang.String))(java.lang.String text): void  Set the title in the window's title bar.  [**showStatusBar**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#showStatusBar(boolean))(boolean show): void  Shows or hides the status bar. |

|  |
| --- |
| GameWindow |
| - backgroundColor: java.awt.Color  - gridColor: java.awt.Color |
| [**GameGrid**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#GameGrid(int, int, int, java.awt.Color, java.lang.String))(int nbHorzCells, int nbVertCells, int cellSize,  java.awt.Color gridColor, java.lang.String bgImagePath)  *Constructs a game window including a playground with a navigation bar, possibly a visible grid and possibly a background image.* |
| +[**setBgColor**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setBgColor(java.awt.Color))(java.awt.Color color): void  Sets the color of the background.  + **setGridColor**(java.awt.Color color): void  Sets the color of the grid and reconstructs the background.  + [**setTitle**](http://www.aplu.ch/classdoc/jgamegrid/ch/aplu/jgamegrid/GameGrid.html#setTitle(java.lang.String))(java.lang.String text): void  Set the title in the window's title bar.  + **refresh**(): void  Refreshs the current game situation (repaint background, tiles, actors).  + **doPause**(): void  Pauses the simulation cycling.  + **doRun**(): void  Starts the simulation cycling. |