

**Package** `jeu_console`

## Class Jeu

`java.lang.Object`  
`jeu_console.Jeu`

```
public class Jeu
extends java.lang.Object
```

The game class It contains the grid and the characters It plays a game on the console

### Constructor Summary

#### Constructors

Constructor	Description
<code>Jeu()</code>	Create a game with default values
<code>Jeu(int height, int width, int nbWalls, int nbHunter, int nbCheater, int nbWiseman, int nbPickaxe, int nbLadder, int nbGlue, int nbRoadMap)</code>	Create a game with a given height, width and number of Walls and Occupants It places all the occupants randomly

### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
<code>Position</code>	<code>getRandomFreePosition()</code>	Gets a random free position in the grid.
<code>Position</code>	<code>getRandomPosition()</code>	Gets a random position in the grid.
<code>void</code>	<code>init()</code>	Initialize the game by adding all the occupants randomly
<code>boolean</code>	<code>isFinished()</code>	Checks if the game is finished.
<code>boolean</code>	<code>isPosFree(Position pos)</code>	Checks if a position is free.

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: [SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

## Methods inherited from class java.lang.Object

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

## Constructor Detail

### Jeu

```
public Jeu(int height,
           int width,
           int nbWalls,
           int nbHunter,
           int nbCheater,
           int nbWiseman,
           int nbPickaxe,
           int nbLadder,
           int nbGlue,
           int nbRoadMap)
```

Create a game with a given height, width and number of Walls and Occupants It places all the occupants randomly

#### Parameters:

height - the height of the grid

width - the width of the grid

nbWalls - the number of walls to add

nbHunter - the number of hunter to add

nbCheater - the number of cheater to add

nbWiseman - the number of wiseman to add

nbPickaxe - the number of pickaxe to add

nbLadder - the number of ladder to add

nbGlue - the number of glue to add

nbRoadMap - the number of road map to add

### Jeu

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: [SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

## Method Detail

### init

```
public void init()
```

Initialize the game by adding all the occupants randomly

**Parameters:**

nbTour - the number of turns to play

### play

```
public void play(int nbTours) throws java.lang.InterruptedException,  
java.io.IOException
```

Plays the game for a given number of turns. It checks if the game is finished.

**Parameters:**

nbTours - the number of turns to play.

**Throws:**

java.lang.InterruptedException - if the thread is interrupted.

java.io.IOException - if an I/O error occurs.

### isFinished

```
public boolean isFinished()
```

Checks if the game is finished. The game is finished if the treasure is found by a hunter.

**Returns:**

true if the game is finished, false otherwise.

### getRandomFreePosition

```
public Position getRandomFreePosition()
```

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

### **getRandomPosition**

```
public Position getRandomPosition()
```

Gets a random position in the grid. It does not create a position with 'Position(height, width)' even if it would give a random position because it could be a position outside the grid (more specifically, it could be on the border of the grid) and it would cause an error for some methods.

**Returns:**

a random position in the grid.

### **isPosFree**

```
public boolean isPosFree(Position pos)
```

Checks if a position is free.

**Parameters:**

pos - the position to check.

**Returns:**

true if the position is free, false otherwise.

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)