

Class Menu

java.lang.Object
Menu

All Implemented Interfaces:
java.awt.event.ActionListener, java.util.EventListener

```
public class Menu
extends java.lang.Object
implements java.awt.event.ActionListener
```

This is the menu class, this will hold our main method and will be used to start the code. In side of this we initialize the menu screen and create a GameJPanel to run the game.

Field Summary

Fields	
Modifier and Type	Field and Description
private static java.awt.Color	backGC This is the creation of the menu for the game.
static javax.swing.JPanel	ButtonPanel
static javax.swing.JPanel	CentralPanel
static javax.swing.JLabel	displayName
static Menu	drawMenu
private java.lang.String	errorName
static java.util.List<javax.swing.JLabel>	highScoreList
static javax.swing.JLabel	highScoreNames
static javax.swing.JFrame	j
static javax.swing.JFrame	menu
private javax.swing.JTextField	name
static javax.swing.JPanel	panel
static java.lang.String	player

<code>static javax.swing.JButton</code>	<code>playGButton</code>
<code>static javax.swing.JButton</code>	<code>quitGButton</code>
<code>static java.awt.Font</code>	<code>RetroGame</code>

Constructor Summary

Constructors

Constructor and Description

`Menu()`

This is the constructor for the menu, when called, it will draw the menu.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
<code>void</code>	<code>actionPerformed(java.awt.event.ActionEvent e)</code>
<code>void</code>	<code>drawMenu()</code> This method is used to draw the menu.
<code>static void</code>	<code>exitGame()</code> This method will end the process
<code>static void</code>	<code>main(java.lang.String[] args)</code> This is the main method of the code and will start the game by creating the menu.
<code>void</code>	<code>startGame(javax.swing.JFrame j)</code> This method will create a instance of the GameJPanel and start the game.

Methods inherited from class `java.lang.Object`

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

Field Detail

`backGC`

`private static final java.awt.Color backGC`

This is the creation of the menu for the game. When you called the menu it will create and JFrame and Panel for the exiting and quitting the game.

CentralPanel

```
public static javax.swing.JPanel CentralPanel
```

panel

```
public static javax.swing.JPanel panel
```

menu

```
public static javax.swing.JFrame menu
```

playGButton

```
public static javax.swing.JButton playGButton
```

quitGButton

```
public static javax.swing.JButton quitGButton
```

ButtonPanel

```
public static javax.swing.JPanel ButtonPanel
```

drawMenu

```
public static Menu drawMenu
```

j

```
public static javax.swing.JFrame j
```

RetroGame

```
public static java.awt.Font RetroGame
```

player

```
public static java.lang.String player
```

highScoreNames

```
public static javax.swing.JLabel highScoreNames
```

highScoreList

```
public static java.util.List<javax.swing.JLabel> highScoreList
```

displayName

```
public static javax.swing.JLabel displayName
```

name

```
private javax.swing.JTextField name
```

errorName

```
private java.lang.String errorName
```

Constructor Detail

Menu

```
public Menu()
```

This is the constructor for the menu, when called, it will draw the menu.

Method Detail

main

```
public static void main(java.lang.String[] args)
```

This is the main method of the code and will start the game by creating the menu.

Parameters:

args - this is unused in our code.

exitGame

```
public static void exitGame()
```

This method will end the process

drawMenu

```
public void drawMenu()
```

This method is used to draw the menu. In this buttons are created and organized in layouts. When this is called you will create a Time FitherX menu.333

startGame

```
public void startGame(javax.swing.JFrame j)
```

This method will create an instance of the GameJPanel and start the game. This will also start a mouse tracer.

Parameters:

j - This is the main frame for graphic to be shown on.

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

actionPerformed in interface java.awt.event.ActionListener

[PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)