

com.charliemouse.cambozola.shared

Class CamStream

java.lang.Object
 java.lang.Thread
 com.charliemouse.cambozola.shared.CamStream

All Implemented Interfaces:

java.lang.Runnable

```
public class CamStream
extends java.lang.Thread
```

com/charliemouse/cambozola/shared/CamStream.java
 Copyright (C) Andy Wilcock, 2001.
 Available from <http://www.charliemouse.com>

This file is part of the Cambozola package (c) Andy Wilcock, 2001.
 Available from <http://www.charliemouse.com>

Cambozola is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Cambozola is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Cambozola; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

A few minor modifications to reduce latency have been made by Kevin Parker These modifications are intended to improve the speed/performance of MJPEG reading (latency has been reduced on my machine by another 10-20 ms; this is per-frame, so there are hundreds of ms of "waiting" reduced away. There are probably more changes that can still be made to greatly improve performance.)

I (Kevin Parker) made several changes to the following code. These include the following. I made the JPEG data stream pipe directly into the parser, rather than loading the whole image into a buffer before parsing. I made a couple of minor changes, one of which was reducing some of the sleep lengths. I also did some minor code auto-cleanup. Finally, I added several comments and javadocs. (I also altered m_retryDelay and removed some unused member variables).

This is not the full version of the Cambozola code, and it has many modifications to make it better suit the needs of a fast-paced FRC tournament. I also changed or removed unneeded pieces of these files.

Nested Class Summary

Nested classes/interfaces inherited from class java.lang.Thread

java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Field Summary

Fields

Modifier and Type	Field and Description
static int	CONNECT_STYLE_HTTP
static int	CONNECT_STYLE_SOCKET

Fields inherited from class java.lang.Thread

MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

Constructors

Constructor and Description
CamStream (java.net.URL strm, java.lang.String app, java.net.URL docBase, int retryCount, int retryDelay, java.util.logging.Logger logger, boolean debug)

Method Summary

Methods

Modifier and Type	Method and Description
void	<code>finalize()</code>
java.awt.image.BufferedImage	<code>getCurrent()</code>
double	<code>getFPS()</code>
int	<code>getIndex()</code>
java.net.URL	<code>getStreamURL()</code>
java.lang.String	<code>getType()</code>
void	<code>run()</code>
void	<code>unhook()</code>

Methods inherited from class java.lang.Thread

[`activeCount`](#), [`checkAccess`](#), [`clone`](#), [`countStackFrames`](#), [`currentThread`](#), [`destroy`](#), [`dumpStack`](#), [`enumerate`](#), [`getAllStackTraces`](#), [`getContextClassLoader`](#), [`getDefaultUncaughtExceptionHandler`](#), [`getId`](#), [`getName`](#), [`getPriority`](#), [`getStackTrace`](#), [`getState`](#), [`getThreadGroup`](#), [`getUncaughtExceptionHandler`](#), [`holdsLock`](#), [`interrupt`](#), [`interrupted`](#), [`isAlive`](#), [`isDaemon`](#), [`isInterrupted`](#), [`join`](#), [`join`](#), [`join`](#), [`resume`](#), [`setContextClassLoader`](#), [`setDaemon`](#), [`setDefaultUncaughtExceptionHandler`](#), [`setName`](#), [`setPriority`](#), [`setUncaughtExceptionHandler`](#), [`sleep`](#), [`sleep`](#), [`start`](#), [`stop`](#), [`stop`](#), [`suspend`](#), [`toString`](#), [`yield`](#)

Methods inherited from class java.lang.Object

[`equals`](#), [`getClass`](#), [`hashCode`](#), [`notify`](#), [`notifyAll`](#), [`wait`](#), [`wait`](#), [`wait`](#)

Field Detail

CONNECT_STYLE_HTTP

```
public static final int CONNECT_STYLE_HTTP
```

See Also:

[Constant Field Values](#)

CONNECT_STYLE_SOCKET

```
public static final int CONNECT_STYLE_SOCKET
```

See Also:

[Constant Field Values](#)

Constructor Detail

CamStream

```
public CamStream(java.net.URL strm,
                 java.lang.String app,
                 java.net.URL docBase,
                 int retryCount,
                 int retryDelay,
                 java.util.logging.Logger logger,
                 boolean debug)
```

Method Detail

finalize

```
public void finalize()
    throws java.lang.Throwable
```

throws `java.lang.Throwable`

Overrides:

finalize in class `java.lang.Object`

Throws:

`java.lang.Throwable`

getCurrent

```
public java.awt.image.BufferedImage getCurrent()
```

getFPS

```
public double getFPS()
```

getIndex

```
public int getIndex()
```

getStreamURL

```
public java.net.URL getStreamURL()
```

getType

```
public java.lang.String getType()
```

run

```
public void run()
```

Specified by:

run in interface `java.lang.Runnable`

Overrides:

run in class `java.lang.Thread`

unhook

```
public void unhook()
```

[Overview](#) [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](#)