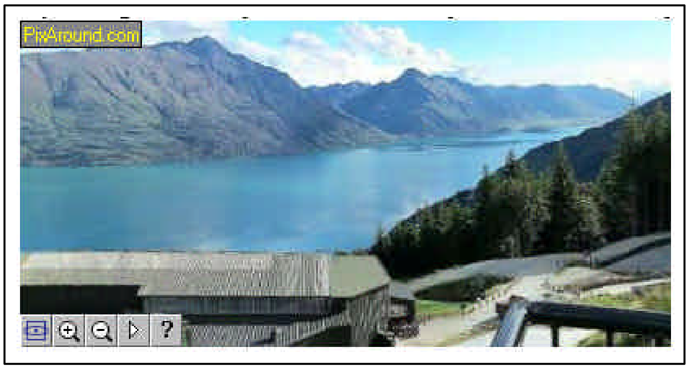
# µPixScreen *Java*

# User Manual

March 2000
Version 1.0.3.0

Every effort has been made to ensure that the information in this manual is accurate. PixAround.com Pte Ltd is not responsible for printing or clerical errors.

## Introduction



µPixScreen.jar ( pronounced as micro-Pix-Screen) Java Applet enables creation of PixAround Webpage. For more information on PixAround Webpage, please visit www.pixaround.com. This document describes the available µPixScreen features and options.

## Features

The fastest way to take a quick tour of µPixScreen's capability is by going through the samples available at www.pixaround.com . µPixScreen includes the following features:

- Runs on Java Virtual machine (JVM) 1.1. Any browser with properly implemented JVM 1.1, e.g. Microsoft Internet Explorer 4.0 and above and Netscape 4.0 and above, should be able to view the applet.
- High quality 24-bit true color rendering with bilinear-filtering.
- Supports both 2D Viewing (very fast), and 3D Viewing (correctly warped).
- Small file size (about 15KB), requiring slightly more than 2 seconds download time using a 56kbps modem.
- Easy to use and configure. Normal usage requires only 4 lines of HTML code.
- Tool panel, help dialog, tooltips provide on-line help.
- Open format. Loads JPEG files.
- Ability to render 360° and non-360° degrees PixAround Scenes.
- Fast rendering. See *Benchmark* .
- Built-in titling capability.
- Supports progressive downloading.
- Highly configurable with more than 15 options to customize µPixScreen applet to your needs. See *Options*.
- Automatically removes PixAround strip.

## Methods

µPixScreen provides the following methods for effective story-telling. You can write a story in html format, then synchronize the story with the µPixScreen Java applet using the following methods. µPixScreen enables loading of a new 360° PixAround Scene, and turning the current view to a specific point of interest.

**TransitTo**(

```
String  strURL,          // URL of new or current PixAround Scene.

double  dSrcYaw,                // source yaw to turn to
double  dSrcPitch,              // source pitch to turn to
double  dSrcFov,                // source field-of-view to turn to

double  dTargetYaw,             // target starting yaw
double  dTargetPitch,           // target starting pitch
double  dTargetFov,             // target starting field-of-view
double  dTotalYawAngle,         // target's total yaw angle


int     iAutoPan,               // Autopan after loading?
double  dSpeed           // turning speed
```

)

Use ***TransitTo*** to load a new PixAround Scene, or to turn to a specific direction in the current PixAround Scene.

Two scenarios can happen:

- If *strURL* is the current PixAround Scene, the current view changes direction gradually to *dTargetYaw*, *dTargetPitch*, and *dTargetFov,* at *dSpeed* velocity*.*
- If *strURL* is a not the current PixAround Scene, the view will first turn to *dSrcYaw*, *dSrcPitch*, and *dSrcFov*, load the *strURL,* and then face *dTargetYaw*, *dTargetPitch*, and *dTargetFov*.

**TransitTo**(

```
String  strURL,
double  dTargetYaw,
double  dTargetPitch,
double  dTargetFov,
double  dTotalYawAngle,
int     iAutoPan,
double  dSpeed)
```

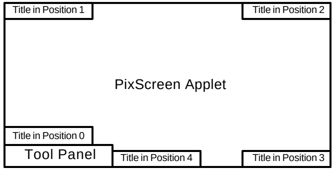
This version of ***TransitTo*** is a convenient function that performs the full TransitTo (see above) ignoring *dSrcYaw*, *dSrcPitch*, and *dSrcFov*. If *strURL* is different from the current PixAround Scene, it loads *strURL* immediately, without turning.

**SetTitle**(String strTitle, int iPos, int foreColor, int backColor, int borderColor)

Sets the title within the applet to *strTitle* at position specified by *iPos (see option for meaning iPos options)*. Colors for text, background, and border color are specified by *foreColor*, *backColor*, and *borderColor* respectively, in aarrggbb hexadecimal format. *iPos*, *foreColor*, *backColor* or *borderColor* will be ignored if set to -1.

# Options

| ***Param*** | ***Description*** | ***Default*** | ***example*** |
|---|---|---|---|
| **code** | Should be uPixScreen.class | NA | *code=**uPixScreen.class*** |
| **archive** | Should be uPixScreen.jar | NA | *archive=**uPixScreen**.jar* |
| **width** | Width of the applet window. | NA | *Width=**400*** |
| **height** | Height of the applet window. | NA | *Height=**200*** |
| **url** | The pathname of PixAround Scene file. This path is relative to where the html document (which contains the applet) resides. | NA | *<param name=url value="**sample.jpg**">* |
| **startYaw** | Starting yaw angle ranging from 0° to 360°. | 0 | Start at 10°.<br>*<param name=**startYaw** value=**10**>* |
| **startPitch** | Starting Pitch angle which ranging from 0° to 89°. | 0 | Start at 5°<br>*<param name=**startPitch** value=**5**>* |
| **startFov** | Starting field of view (FOV). The range on FOV is subject to the PixAround Scene to be viewed. Any value that goes out of range will be clipped. Hint: set this to 180° for max FOV. | Maximum FOV - 10° | For maximum FOV.<br>*<param name=**startFov** value=**180**>* |
| **partialStitch** | Indicate if the PixAround Scene is an image that is less than 360°. The values can be "yes" or "no". If this value is "no" but ***horFov*** is not 360°, the viewer will still treat the PixAround Scene as a partial image. | *no* | One-liner parameter for non 360 degrees PixAround Scene. The applet estimates the horFov.<br>*<param name=partialStitch value=yes>* |
| **horFov** | Indicate the field of view of the PixAround Scene. The value can range from 0 to 360°. If this value is omitted but ***partialStitch*** is "yes", the viewer will automatically estimate the value of ***horFo***v. | 360 | Another one-liner for a 180 degrees PixAround Scene.<br>*<param name=horFov value=180>* |
| **autoPan** | Option to pan automatically when first downloaded. The values can be "yes" or "no". | yes | To turn off autoPan,<br>*<param name=horFov value=no>* |
| **autoPanSpeed** | Speed of AutoPan | 10 | Slower autoPan,<br>*<param name=autoPanSpeed value=2>* |
| **autoPanDelay** | µPixScreen politely sleeps for ***autoPanDelay*** milliseconds between frames, giving the rest of the html objects computational resources.<br><br>*hint*:<br>If you are confident that no other objects in the HTML requires computational resources, you can safely set this value to 0, ensuring a smoother auto pan. In our experience, you can safely set this to 0 for Internet Explorer.<br><br>*Warning*: Setting to a lower value may cause Netscape window event loss, resulting in improper Netscape repainting (e.g. Netscape button s becomes black). | 50 | Smoother (more selfish) autoPan,<br>*<param name=autoPanDelay value=0>* |

| ***Param*** | ***Description*** | ***Default*** | ***example*** |
|---|---|---|---|
| **autoPanTimeout** | Stop auto panning after a time specified by ***autoPanTimeout*** (in minutes).<br><br>*Note*: Prolonged autopan is known to cause Netscape to lose windows event. ***autoPanTimeout*** is created to prevent this Netscape problem. | 3 | *Time out after 5 minutes.*<br>*<param name=autoPanTimeout value=5>* |
| **maxFps** | Maximum frames per second. Should be less than or equal to 60.<br><br>*Note*: Frame rate above 60 Hz is imperceptible by most people, and so it's generally pointless to be above 60 Hz. Most people perceive smooth motion at 30 Hz . A fast machine ( e.g. Pentium III 450 Mhz) is able to produce roughly 55 Hz. By clamping the frame rate to 30 Hz , we relinquish some CPU resources, resulting in the following benefits (at the cost of appearing slightly jerky):<br>• Netscape in Linux responses better to mouse event.<br>• Other applets/html objects have more CPU resources. | 60 | *<param name=maxFps value=30>* |
| **hq** | 0 - Default normal quality viewing.<br>1 - Default high quality viewing.<br><br>*Note*: High quality viewing has no effect on 2D view.<br>*Warning*: Setting the default viewing to high quality may result in slow performance on slower computer (e.g. Pentium 100 Mhz). | 0 | *<param name=hq value=1>* |
| **viewType** | Default viewing type.<br>1 - 3D view<br>2 - 2D view<br><br>*Hint*: 2D view is very fast, even on a 486 computer! In addition, 2D view does not shimmer during panning. From our experience, some people cannot tell the difference between 3D view and 2D view (unless we show them side by side). 2D view works ok for outdoor scene. Thus, you may want to default to 2D view for certain scenes. You judge it yourself. | 1 | *Start with 2D View.*<br>*<param name=viewType value=2>* |
| **title** | Show a title. If title is "", then title will not be displayed. | "" | *<param name=title value="My room">* |
| **titlePos** | Title position.<br> Title in Position 1<br>Title in Position 2<br>PixScreen Applet<br>Title in Position 0<br>Tool Panel<br>Title in Position 4<br>Title in Position 3 | 0 | *Place title on the right bottom corner.*<br>*<param name=titlePos value=3>* |

| ***Param*** | ***Description*** | ***Default*** | ***example*** |
|---|---|---|---|
| **titleForegroundColor** | Color of title text in hexadecimal. aarrggbb. | ffffff | Yellow foreground,<br>&lt;param name=titleForegroundColor value =ffff00&gt; |
| **titleBackgroundColor** | Color of title background in hexadecimal | 888888 | |
| **titleBorderColor** | Color of title Border in hexadecimal. Hint: set this to background color to turn off border. | 666666 | |

## Benchmark

More than 50 frames per second on a Pentium III 450, 64 MB RAM, ATI 3D Rage Pro, Applet Size 400 x 200 pixels, running on 24bpp, using Microsoft Internet Explorer 5.0. More benchmarks to come.

## Contacts

support@pixaround.com

PixAround.com Pte Ltd