

# Visionneuse PanoramaStudio 2.0

## Documentation

---

© 2006-2008 Tobias Hüllmandel Softwareentwicklung  
<http://www.tshsoft.com/fr>

## Table des matières

<b>1 La Visionneuse PanoramaStudio .....</b>	<b>1</b>
1.1 Description et configuration requise .....	1
1.2 Fonctionnalités .....	1
1.3 Intégration dans PanoramaStudio .....	1
1.4 Utilisation de la visionneuse avec des panoramas issus d'autres logiciels.....	2
1.5 Pilotage de la Visionneuse PanoramaStudio.....	2
1.6 Licence .....	2
<b>2 Inclure manuellement la visionneuse dans des pages HTML.....</b>	<b>4</b>
2.1 Version Java : la balise <applet> .....	4
2.2 Version Flash : les balises <object> et <embed> .....	4
2.3 Fichiers requis pour l'utilisation de l'applet .....	5
2.4 Pilotage à l'aide de JavaScript .....	5
<b>3 Description des paramètres.....</b>	<b>6</b>
3.1 Les paramètres de base.....	7
3.2 Paramètres de lecture automatique.....	9
3.3 Paramètres de la barre de navigation.....	10
3.4 Paramètres des hotspots .....	11
<b>4 Hotspots et visites virtuelles .....</b>	<b>11</b>
<b>5 Exemple.....</b>	<b>12</b>

# 1 La Visionneuse PanoramaStudio

## 1.1 Description et configuration requise

La *Visionneuse PanoramaStudio* permet d'afficher de manière interactive les panoramas sphériques ou cylindriques générés par PanoramaStudio.

La version 2.0 de la *Visionneuse PanoramaStudio* est à présent déclinée en une version Java et en une version Flash.

La version Java nécessite une machine virtuelle Java version 1.1 et supérieur sous un environnement compatible Java pour fonctionner (Windows, Linux, MacOS, ...). Le visionnage quant à lui ne nécessite que le plugin Java, dans la mesure où l'espace mémoire disponible pour afficher le panorama est suffisant.

La version Flash nécessite le lecteur Adobe Flash Player à partir de la version 9, et, dans la mesure où l'espace mémoire disponible pour afficher le panorama est suffisant, les panoramas devraient s'afficher sans problème dans tous les environnements compatibles avec ce lecteur.

Les deux déclinaisons de la *Visionneuse PanoramaStudio* s'utilisent et se paramètrent exactement de la même manière ; il est ainsi très aisé de basculer entre les deux.

Les principaux avantages de la visionneuse sont la vitesse d'affichage des panoramas interactifs tout en assurant la conformité de l'affichage dans le respect de la perspective, l'intégration dans PanoramaStudio ainsi que la possibilité d'intégrer des hotspots en vue de créer des visites virtuelles complètes.

## 1.2 Fonctionnalités

- Présentation des panoramas interactive et indépendante du système d'exploitation
- Affichage de tout panorama jusqu'aux panoramas sphériques complets 360° x 180°
- Code Java et Flash hautement optimisé pour un affichage rapide
- Support de la projection sphérique et cylindrique
- Pilotage de la visionneuse en JavaScript
- Utilisation de hotspots et possibilité de créer des visites virtuelles
- Adaptation dynamique du niveau de qualité du panorama en fonction de la puissance de calcul disponible
- Lecture automatique ; possibilité de travelling horizontal et vertical, zoom
- Utilisation du clavier et de la molette de la souris pour se déplacer dans le panorama

## 1.3 Intégration dans PanoramaStudio

La *Visionneuse PanoramaStudio* est intégrée à PanoramaStudio à compter de la version 1.6. Ceci permet de créer directement et automatiquement des pages Web incluant un panorama depuis PanoramaStudio et de générer le code HTML ainsi que les fichiers nécessaires selon le paramétrage effectué. Le code généré prend en charge la majorité du paramétrage et des fonctionnalités décrits dans cette documentation.

## 1.4 Utilisation de la visionneuse avec des panoramas issus d'autres logiciels

En principe, il est possible d'utiliser la visionneuse pour visualiser des panoramas cylindriques et sphériques créés avec d'autres applications que PanoramaStudio. Les images doivent se trouver au format JPEG. La documentation ci-dessous décrit comment intégrer la visionneuse dans des pages Web existantes.

## 1.5 Pilotage de la Visionneuse PanoramaStudio

La *Visionneuse PanoramaStudio* permet de naviguer de manière interactive dans un panorama. Cela se fait, dans un premier temps, à l'aide de la souris. Appuyez sur le bouton gauche de la souris et déplacez-la pour vous déplacer latéralement et verticalement dans le panorama. Vous pouvez également utiliser la molette de la souris pour zoomer/dézoomer (version de Java 1.4 et plus). Si la barre de navigation est affichée, vous pouvez utiliser les fonctions pour zoomer ou naviguer dans le panorama, pour démarrer ou arrêter la lecture automatique ou pour afficher ou masquer les hotspots.

La visionneuse se pilote également grâce au clavier. Les touches du pavé fléché permettent de naviguer dans le panorama. '+' et '-' vous permettent de zoomer ou de dézoomer. La barre espace permet de démarrer et arrêter la lecture automatique. 'h' vous permet d'afficher ou de masquer les hotspots, 't' permet de masquer ou afficher la barre de navigation.

## 1.6 Licence

### Contrat de licence

Visionneuse PanoramaStudio - Version 2.0  
pour machine virtuelle Java à partir de la version 1.1  
et Adobe Flash Player à partir de la version 9  
Copyright © 2005-2008, Tobias Hüllmandel Softwareentwicklung  
Internet : <http://www.tshsoft.com/fr>  
E-Mail : [support-fr@tshsoft.com](mailto:support-fr@tshsoft.com)

1. LICENCE : L'auteur (Tobias Hüllmandel Softwareentwicklung) vous concède en vertu du présent contrat une licence du logiciel "Visionneuse PanoramaStudio 2.0" ("logiciel"), à l'unique condition que vous acceptiez l'ensemble des dispositions contenues dans le présent contrat de licence. Le logiciel comprend une documentation électronique ainsi que le contrat de licence ("Licence.txt").
2. UTILISATION COMMERCIALE/PRIVEE
  - 2.1 UTILISATION PRIVEE  
L'utilisation privée, non commerciale du logiciel, même sur Internet, est gratuite et ne nécessite pas de licence supplémentaire.
  - 2.2 UTILISATION COMMERCIALE  
Pour une utilisation commerciale sur Internet, vous devez acquérir une licence pour chacune de vos présences sur Internet au plus tard à l'échéance de la période d'essai de 30 jours.  
Une présence sur Internet est caractérisée par le nom de domaine du site Internet (exemple "www.tshsoft.com"). Une licence est valable pour un nom de domaine et pour tous les alias qui pointent sur ce même nom de domaine. Le nombre de panoramas pouvant être affichés sur cette présence Internet n'est pas limité.  
L'utilisation commerciale sur Internet se caractérise par le fait que l'éditeur du dit site en fait un usage commercial, qu'il n'est pas un particulier, une association ou une institution académique.
3. ACTIONS NON AUTORISEES
  - 3.1 Utilisation du logiciel sous une forme non autorisée par le présent contrat.

3.2 L'utilisation commerciale d'une copie non enregistrée du logiciel après l'échéance de la période d'essai de 30 jours.

3.3 Toute forme d'ingénierie inverse (reverse engineering), de décompilation et de désassemblage du logiciel.

3.4 Toute modification du logiciel.

4. Dans les limites autorisées par la loi, l'auteur ne pourra en aucun cas être responsables de toutes pertes de revenus, pertes de profits ou pertes de données, ni de tout dommage spécial, incident, consécutif, indirect ou punitif, quelle que soit la cause et le fondement de la responsabilité, résultant de l'utilisation ou de l'impossibilité d'utiliser le logiciel, y compris lorsque l'auteur avait connaissance de l'éventualité de tels dommages.

### Obtenir une licence

Conformément aux termes du contrat de licence ci-dessus, l'utilisation commerciale sur Internet de la *Visionneuse PanoramaStudio* nécessite l'obtention d'une licence au plus tard après une période d'essai de 30 jours auprès du service d'enregistrement ShareIt édité par la société Digital River GmbH. L'achat d'une licence vous permet d'obtenir une clé d'enregistrement correspondant à votre nom de domaine.

### Cette clé d'enregistrement vous permet :

- d'obtenir le droit d'utiliser la *VisionneusePanoramaStudio* sur un site Web à vocation commerciale. La licence vous permet d'utiliser la visionneuse pour un nombre illimité de panoramas sur le site correspondant.
- la possibilité d'inclure votre logo dans la visionneuse,
- de bénéficier d'une assistance gratuite en cas de questions ou de problèmes.

### Acquérir une clé d'enregistrement

L'acquisition d'une clé d'enregistrement s'effectue de manière simple et sécurisée sur Internet grâce au service ShareIt. ShareIt permet les règlements par carte de crédit, PayPal, chèque, virement bancaire ou espèces. Vous obtiendrez votre clé d'enregistrement personnelle par email dans les 24 heures suivant votre paiement. Si vous réglez par carte de crédit, la clé d'enregistrement est généralement envoyée dans la minute suivant le paiement.

Pour commander, connectez-vous sur :

<http://www.tshsoft.com/fr/panoramastudio/register.html>

### Assistance

Si vous avez des questions sur l'enregistrement, le paiement ou la livraison, connectez-vous au Service Clientèle de ShareIt :

<http://shareit1.element5.de/ccc/index.html?publisherid=20959&languageid=6>

## 2 Inclure manuellement la visionneuse dans des pages HTML

### 2.1 Version Java : la balise <applet>

La balise <applet> permet d'inclure la *Visionneuse PanoramaStudio* dans une page Web. L'applet Java correspondante est constituée du fichier *panoStudioViewer.jar*. L'exemple suivant est une illustration simple de l'utilisation de la balise <applet> :

```
<applet archive="panoStudioViewer.jar" code="panoStudioViewer.PanoStudioViewer.class"
  name="MonPanorama" width="800" height="480">
  <param name="pano" value="MonPanorama.xml">
</applet>
```

Tous les paramètres permettant de décrire et configurer le panorama sont situés dans le fichier de paramétrage XML désigné par le paramètre `param name='pano'`. Dans l'exemple ci-dessus, ce paramètre désigne le fichier de paramétrage *MonPanorama.xml*. Une description complète du fichier de paramétrage XML se trouve au chapitre 3 (p. 6).

### 2.2 Version Flash : les balises <object> et <embed>

L'insertion d'un panorama Flash dans des pages Web s'appuie sur la version Flash de la *Visionneuse PanoramaStudio*. Celle-ci est constituée du fichier *panoStudioViewer.swf*. La mise en œuvre d'un panorama Flash nécessite plus d'effort que la mise en œuvre d'un panorama Java. L'exemple suivant est une illustration de l'utilisation de la visionneuse Flash :

```
<object classid="CLSID:D27CDB6E-AE6D-11cf-96B8-444553540000" width="800"
  height="480" id="MonPanorama"
  codebase="http://active.macromedia.com/flash9/cabs/swflash.cab#version=9,0,28,0">
  <param name="movie" value="panoStudioViewer.swf" />
  <param name="allowScriptAccess" value="always" />
  <param name="allowNetworking" value="all" />
  <param name="allowFullScreen" value="true" />
  <param name="FlashVars" value="pano=MonPanorama.xml" />
  <embed src="panoStudioViewer.swf" width="800" height="480"
    type="application/x-shockwave-flash" name="MonPanorama"
    allowScriptAccess="always" allowNetworking="all" allowFullScreen="true"
    FlashVars="pano=MonPanorama.xml"
    pluginspage="http://www.macromedia.com/go/getflashplayer">
  </embed>
</object>
```

De même que pour la version Java, tous les paramètres permettant de décrire et configurer le panorama sont situés dans le fichier de paramétrage XML *MonPanorama.xml*. Le fichier XML doit pour cela être référencé au sein des balises <object> et <embed> dans le paramètre `FlashVars`, tel qu'illustré dans l'exemple.

Un des principaux avantages est que le fichier de paramétrage XML est commun aux déclinaisons Flash et Java de la *Visionneuse PanoramaStudio*. Ainsi, un panorama peut être affiché indifféremment par l'une des deux versions sans qu'il ne soit nécessaire de modifier le fichier de paramétrage XML ou l'image du panorama.

Lors de l'adaptation du code ci-dessus dans une page Web, il faut s'assurer de la cohérence des valeurs des paramètres `width` et `height` dans les balises <object> et <embed>. De même, la valeur du paramètre

tre id de la balise `<object>` devra toujours être identique à la valeur du paramètre name de la balise `<embed>`.

## 2.3 Fichiers requis pour l'utilisation de l'applet

Pour publier une page Web contenant un panorama, vous devez disposer d'un ensemble de fichiers en plus du fichier HTML. Il s'agit principalement de l'applet Java contenue dans le fichier `panoStudioViewer.jar` ou de l'applet Flash contenue dans le fichier `panoStudioViewer.swf`, du fichier de paramétrage XML permettant de décrire et configurer le panorama ainsi que du fichier JPEG contenant l'image de panorama.

Lors de la création de panoramas Flash ou Java, il faut prendre en compte les limitations de taille d'image des panoramas. Dans le cas de panoramas Java, cette limite est fixée par la quantité de mémoire vive allouée à l'applet par le navigateur. En règle générale, mieux vaut ne pas dépasser une taille d'image de 15 ou 20 mégapixels. Dans le cas de panoramas Flash, la limite de taille se situe à une image de 8192 pixels de côtés. Cette limite est imposée par la plupart des versions du lecteur Flash et ce dernier refusera de charger des images dont la taille excède la limite citée ci-dessus.

Dans le cas où vous utiliser des hotspots, le fichier correspondant doit également être publié. Voir aussi le chapitre 4 (p. 11).

Essayez si possible de copier tous les fichiers dans le même répertoire. Dans le cas contraire, vous devrez spécifier les chemins des fichiers dans le code HTML.

## 2.4 Pilotage à l'aide de JavaScript

Il est possible de piloter les deux déclinaisons de la *Visionneuse PanoramaStudio* à l'aide de scripts JavaScript au sein d'une page Web.

**Attention !** Dans le cas de la version Flash, il faut noter que, pour des raisons de sécurité imposées par le lecteur Flash, le pilotage ne pourra pas se faire depuis des scripts JavaScript situés en local, c'est-à-dire que les fonctions JavaScripts ne fonctionneront qu'une fois ligne sur une page Web hébergée sur un serveur Web.

Pour le pilotage à l'aide de JavaScript, les fonctions suivantes sont implémentées :

- `void startAutoRotate()`  
Démarre la lecture automatique.
- `void stopAutoRotate()`  
Arrête la lecture automatique.
- `void toggleHotspots()`  
Affiche ou masque les hotspots.
- `void setAutoRotate(double autoPanRate, double autoTiltRate, double autoZoomRate)`  
Spécifie les paramètres de la lecture automatique. Les paramètres de la fonction sont décrits au chapitre 3.2 (p. 9).
- `void showToolbar()`  
Affiche la barre de navigation dans la fenêtre de l'applet.
- `void hideToolbar()`  
Masque la barre de navigation.

- `void setView(double pan, double tilt, double hfov)`

Positionne le panorama selon l'angle horizontal `pan`, l'angle vertical `tilt` et le champ de vision à l'angle `hfov`.

- `boolean isRotating()`

Indique si la lecture automatique est activée.

- `void openPanorama(String filename)`

Ouvre un autre panorama dans la fenêtre de l'applet. `filename` doit contenir le nom du fichier de paramétrage XML correspondant au panorama.

### Exemples JavaScript

Voici deux exemples de code JavaScript pour créer un bouton *Lecture* et un bouton *Pause* pour commander la lecture automatique. Le nom de l'objet `<applet>` ou bien `<object>` et `<embed>` est, comme dans les exemples des chapitres 2.1 (p. 4) et 2.2 (p. 4), `MonPanorama`.

```
<INPUT type="button" value="Lecture"
onClick="document.MonPanorama.startAutoRotate();">
<INPUT type="button" value="Arrêt"
onClick="document.MonPanorama.stopAutoRotate();">
```

Voici un exemple plus complexe n'utilisant qu'un seul bouton combiné Lecture/Arrêt :

```
<INPUT type="button" value="Lecture/Arrêt" onClick="
if (document.MonPanorama.isRotating())
    document.MonPanorama.stopAutoRotate();
else document.MonPanorama.startAutoRotate();
">
```

Voici un exemple de lien comportant une action JavaScript. Ce lien permet d'inverser le sens de rotation d'un panorama :

```
<a href=# onClick="document.MonPanorama.setAutoRotate(-2.0,-1.0, 1.0);">Inverser
la rotation</a>
```

## 3 Description des paramètres

Tous les paramètres permettant de décrire et de configurer le panorama se situent au sein du fichier de paramétrage XML.

Le fichier de paramétrage XML est constitué d'un en-tête, d'un élément `<panoramaStudioViewer>` et d'un élément subordonné `<panorama>`. L'en-tête XML comporte la version XML ainsi que le type d'encodage du jeu de caractères. Celui-ci doit systématiquement être identique à :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

La structure générale d'un fichier de paramétrage XML se construit comme suit :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<panoramaStudioViewer>
  <panorama>
    [...]
  </panorama>
</panoramaStudioViewer>
```

Tout le paramétrage s'effectue au sein de la balise <panorama>. Chaque paramètre constitue un élément XML composé d'une balise marquant le début et d'une balise marquant la fin de cet élément. L'exemple ci-dessous illustre l'utilisation du paramètre `file` :

```
<file>Panorama.jpg</file>
```

La visionneuse permet d'utiliser les paramètres suivants :

### 3.1 Les paramètres de base

- `file`  
*Nom de fichier/URL*  
Nom de fichier du panorama. Le fichier doit être au format JPEG.
- `hsfile`  
*Nom de fichier/URL*  
Nom de fichier du fichier hotspots. Le fichier doit être au format GIF avec une profondeur de couleurs de 8 bits. Voir aussi le chapitre 4 (p. 11).
- `key`  
*String*  
Paramètre contenant la clé de licence. Celle-ci doit figurer sur les pages Web commerciales. Référez-vous également au contrat de licence.
- `lang`  
*String*  
Paramètre de la langue d'affichage des messages dans la visionneuse. Il peut prendre les valeurs `fr` (français), `en` (anglais) ou `de` (allemand).
- `pan`  
*Float*  
Angle de vision horizontal initial.
- `tilt`  
*Float*  
Angle de vision horizontal initial.
- `hfov`  
*Float*  
Champ de vision initial.
- `minpan`  
*Float ; valeur par défaut : 0.0*  
Angle de vision horizontal minimal ; c'est l'angle qui définit le bord gauche du panorama.
- `maxpan`  
*Float ; valeur par défaut : 360.0*  
Angle de vision horizontal maximal ; c'est l'angle qui définit le bord droit du panorama.
- `mintilt`  
*Float*  
Angle de vision vertical minimal ; c'est l'angle qui définit le bord supérieur du panorama.



- `maxtilt`  
*Float*  
Angle de vision vertical maximal ; c'est l'angle qui définit le bord inférieur du panorama.
- `minhfov`  
*Float ; [5.0;165.0] ; valeur par défaut : 5.0*  
Champ de vision horizontal minimal. Cette valeur fixe la limite du zoom dans l'image.
- `maxhfov`  
*Float ; [5.0;165.0] ; valeur par défaut : 5.0*  
Champ de vision horizontal maximal. Cette valeur fixe la limite du dézoom dans l'image.
- `spherical`  
*Boolean ; true/false ; valeur par défaut : true*  
Panorama sphérique ou cylindrique.
- `quality`  
*Integer ; [0;4] ; valeur par défaut : 1*  
Réglage du niveau de qualité d'affichage.  
Niveaux : 0 : faible/rapide, 1 : faible, 2 : moyen, 3 : élevé, 4 : dynamique.  
En mode qualité élevée, une interpolation bilinéaire est effectuée. En position 0, l'interpolation est toujours désactivée. L'interpolation est activée successivement sur des images fixes, la lecture automatique et la navigation manuelle sur les niveaux supérieurs. Le niveau 4 (dynamique) permet d'adapter le niveau de qualité en fonction de la puissance de calcul disponible.
- `showToolbar`  
*Boolean ; true/false ; valeur par défaut : true*  
Affichage la barre de navigation.
- `waitImage`  
*Nom de fichier/URL*  
Optionnel. Image affichée pendant le chargement du panorama sur l'écran de chargement. L'image doit être au format PNG, JPEG ou GIF.
- `logo`  
*Nom de fichier/URL*  
Nom de fichier ou line vers un logo qui est affiché en place du logo PanoramaStudio. Cette option n'est active que si l'applet tourne en local sur une machine ou si le paramètre `key` comporte une licence valide.
- `logoUrl`  
*Nom de fichier/URL*  
URL liée au logo personnalisé, si disponible.
- `logoX` et `logoY`  
*Integer ; valeur par défaut : 4*  
Position du logo personnalisé en partant de l'angle en haut à droite.
- `bgWait`  
*Couleur RGB ; hexadécimal ; valeur par défaut : 000000*  
Couleur de fond de l'écran de chargement.

- `fgWait`  
*Couleur RGB ; hexadécimal ; valeur par défaut : FFFFFFFF*  
Couleur du premier plan de l'écran de chargement.
- `loadString`  
*String ; valeur par défaut : « Chargement : »*  
Message de l'écran de chargement à proximité de la barre de progression.
- `title`  
*String*  
Titre du panorama. Le titre du panorama est affiché dans la barre d'état.
- `showBorder`  
*Boolean ; true/false*  
Affichage d'un cadre autour du panorama.
- `borderColor`  
*Couleur RGB ; hexadécimal ; valeur par défaut : 000000*  
Couleur du cadre de la fenêtre d'applet.
- `textColor`  
*Couleur RGB ; hexadécimal ; valeur par défaut : FFFFFFFF*  
Couleur du texte dans la barre d'état de l'applet.
- `shadowColor`  
*Couleur RGB ; hexadécimal ; valeur par défaut : 000000*  
Couleur de l'ombre du texte de la barre d'état.
- `showHotspots`  
*Boolean ; true/false*  
Affichage des hotspots.
- `mass`  
*Float ; [0.1;2.0] ; valeur par défaut : 0.5*  
Permet de régler la valeur de l'inertie du déplacement dans le panorama, c'est-à-dire le fait que les mouvements démarrent et s'arrêtent progressivement.
- `mouseSen`  
*Float ; [0.1;2.0] ; valeur par défaut : 1.2*  
Permet de régler la sensibilité de la souris. Au plus la valeur est importante, au plus un déplacement dans le panorama se fera rapidement.

## 3.2 Paramètres de lecture automatique

Les paramètres qui s'appliquent à la lecture automatique doivent se situer au sein de l'élément `<autoplay>` :

- `autoRotate`  
*Boolean ; true/false ; valeur par défaut : false*  
Activation la lecture automatique.
- `autoPanRate`  
*Float ; valeur par défaut : 2.0*  
Vitesse de rotation horizontale lors de la lecture automatique (en °/sec).

- `autoTiltRate`  
*Float ; valeur par défaut : 0.0*  
Vitesse de rotation verticale lors de la lecture automatique (en °/sec).
- `autoZoomRate`  
*Float ; valeur par défaut : 1.0*  
Variation du zoom lors de la lecture automatique. 1.0 n'effectue aucune variation du zoom. Les valeurs inférieures à 1.0 permettent de zoomer, des valeurs supérieures à 1.0 de dézoomer.
- `autoRotateRestart`  
*Integer ; valeur par défaut : 5*  
Délai en secondes avant de redémarrer la lecture automatique après une action de l'utilisateur.

### 3.3 Paramètres de la barre de navigation

Les paramètres qui s'appliquent à la configuration de la barre de navigation doivent se situer au sein de l'élément `<toolbar>` :

- `tbLayout`  
*Integer ; [0;2]; valeur par défaut : 0*  
Permet de sélectionner le style de la barre de navigation parmi les trois styles prédéfinis.
- `tbPosition`  
*Integer ; [0;1]; valeur par défaut : 0*  
Position de la barre de navigation dans la fenêtre de l'applet (0 : en bas, 1 : en haut).
- `tbButtonType`  
*Integer ; [0;1]; valeur par défaut : 0*  
Type de bouton de la barre de navigation. Le type 0 correspond à des boutons aux coins arrondis, type 1 correspond à des boutons aux angles droits.
- `tbColorProfile`  
*Integer ; [0;4]; valeur par défaut : 0*  
Permet de sélectionner l'un des cinq modèles de couleur de la barre de navigation
- `tbShowBackground`  
*Boolean ; true/false*  
Affiche un fond sous la barre de navigation.
- `tbBGOpacity`  
*Integer ; [0;255] ; valeur par défaut : 128*  
Transparence du fond de la barre de navigation. 255 correspond à un fond opaque, 0 à un fond invisible.
- `tbButtonOpacity`  
*Integer ; [0;255] ; valeur par défaut : 192*  
Transparence des boutons.
- `tbHotButtonOpacity`  
*Integer ; [0;255] ; valeur par défaut : 255*  
Transparence du bouton activé. Un bouton est activé lorsqu'il est survolé par le curseur de la souris.

- `tbShowPlayButton`

*Boolean ; true/false*

Affichage du bouton de lecture automatique qui permet de démarrer/arrêter la lecture automatique.

- `tbButtonWidth`

*Integer ; [18;48] ; valeur par défaut : 24*

Largeur des boutons de la barre de navigation.

- `tbButtonHeight`

*Integer ; [18;48] ; valeur par défaut : 20*

Hauteur des boutons de la barre de navigation.

### 3.4 Paramètres des hotspots

Il est possible d'utiliser jusqu'à 255 hotspots. Les hotspots sont décrits au sein d'un élément `<hotspotN>`, où N est une valeur qui se situe entre 0 et 254. Au sein de l'élément `<hotspotN>`, on dispose des paramètres suivants :

- `url`

URL à atteindre lorsqu'on clique sur le hotspot.

- `target`

*String*

Définition de la cible (*target*) du lien HTML du hotspot. Il s'agit d'un principe issu du code HTML et permet de spécifier la cible dans laquelle doit s'ouvrir le lien (par exemple `_blank` pour ouvrir dans une nouvelle fenêtre). Dans le contexte de la *Visionneuse PanoramaStudio*, `myself` constitue un cas particulier et permet d'ouvrir un panorama dans l'encart de la visionneuse de la page Web en cours. Dans ce cas, l'URL doit pointer sur le fichier de paramétrage XML du panorama à ouvrir.

- `comment`

*String*

Commentaire pour le hotspot. Le commentaire s'affiche dans la barre d'état de la visionneuse quand le curseur de la souris survole le hotspot.

- `hscolor`

*Couleur RGB ; hexadécimal*

Couleur du hotspot au format hexadécimal.

## 4 Hotspots et visites virtuelles

Comme décrit dans le chapitre *Paramètres des hotspots* ci-dessus (chap. 3.4, p. 11), la *Visionneuse PanoramaStudio* permet de gérer les hotspots. Les hotspots permettent d'insérer des liens vers des pages Web ou bien vers d'autres panoramas. Un hotspot définit une zone dans le panorama qui comporte un lien qui est activé lorsqu'on clique dessus.

Il est ainsi possible de créer des visites virtuelles à travers plusieurs salles d'un bâtiment par exemple dans lequel le visiteur se déplace de manière interactive.

La position et l'aspect des hotspots à destination de la *Visionneuse PanoramaStudio* sont, depuis la version 2.0, enregistrés dans un fichier image au format GIF 8 bit dans lequel chaque hotspot possède

une couleur lui étant propre. Cette image doit avoir exactement la même taille que l'image du panorama. En prenant en compte le fond de l'image, ceci limite le nombre de hotspots à 255 par panorama.

Dans l'image correspondant aux hotspots, le code couleur 0 correspond au fond, le premier hotspot a comme couleur le code couleur 1, et ainsi de suite. Les liens et les autres propriétés du premier hotspot se configurent grâce aux paramètres `url`, `target`, `hscolor` et `comment` de l'élément `hotspot0` (voir le chapitre 3.4, p. 11). Les autres hotspots se définissent de manière analogue.

Le lien avec le fichier définissant les hotspots se fait dans le fichier de paramétrage XML grâce au paramètre `hsfile`.

## 5 Exemple

L'exemple suivant décrit un fichier de paramétrage XML. Le fichier « `panorama_1.jpg` » constitue l'image du panorama, le fichier « `panorama_1.gif` » contient les hotspots correspondants. On voit dans l'exemple que le champ de vision, entre autres paramètres, vaut  $236,51^\circ$  et qu'il s'agit d'un panorama cylindrique. Le panorama comporte un hotspot (`hotspot0`) qui pointe sur un autre panorama « `panorama_2.xml` ». Ce panorama s'afficherait au sein de la même page en raison de la cible définie grâce au mot clé `myself`.

```
<?xmlversion="1.0" encoding="UTF-8" ?>
<panoramaStudioViewer>
  <panorama>
    <file>panorama_1.jpg</file>
    <hsfile>panorama_1.gif</hsfile>
    <mintilt>-41.53</mintilt>
    <maxtilt>41.53</maxtilt>
    <minpan>0.00</minpan>
    <maxpan>236.51</maxpan>
    <pan>118.26</pan>
    <tilt>0.00</tilt>
    <hfov>70.00</hfov>
    <spherical>>false</spherical>
    <mouseSen>1.00</mouseSen>
    <loadString>Loading:</loadString>
    <title></title>
    <quality>4</quality>
    <autoplay>
      <autoPanRate>3.0</autoPanRate>
    </autoplay>
    <toolbar>
      <showToolbar>true</showToolbar>
    </toolbar>
    <hotspot0>
      <url>panorama_2.xml</url>
      <target>myself</target>
    </hotspot0>
  </panorama>
</panoramaStudioViewer>
```

Pour inclure ce panorama « `panorama_1.xml` » dans une page Web au format Java, il faudrait insérer le code HTML suivant :

```
<applet archive="panoStudioViewer.jar" code="panoStudioViewer.PanoStudioViewer.class"
  name="pano" width="600" height="400">
  <param name="pano" value="panorama_1.xml">
</applet>
```

Si l'on souhaitait afficher le même panorama à l'aide de la visionneuse Flash, on utiliserait le code HTML suivant :

```
<object classid="CLSID:D27CDB6E-AE6D-11cf-96B8-444553540000" width="600" height="400"
id="pano"
codebase="http://active.macromedia.com/flash9/cabs/swflash.cab#version=9,0,28,0">
  <param name="movie" value="panoStudioViewer.swf" />
  <param name="allowScriptAccess" value="always" />
  <param name="allowNetworking" value="all" />
  <param name="allowFullScreen" value="true" />
  <param name="FlashVars" value="pano=panorama_1.xml" />
  <embed src="panoStudioViewer.swf" width="600" height="400"
type="application/xshockwave-flash" name="pano" allowScriptAccess="always"
allowNetworking="all" allowFullScreen="true" FlashVars="pano=panorama_1.xml"
pluginspage="http://www.macromedia.com/go/getflashplayer" >
  </embed>
</object>
```