

# HOMEWORLD2

## Sensors Manager

This document describes the various features of the HW2 sensors manager.

### Script files

The following data files are primarily Sensors Manager-related:

File	Description
UI\SensorsManager\SensorsManager.lua	Miscellaneous Sensors Manager tweaks and world graphics ATI template
UI\SensorsManager\Blobs.lua	ATI templates for rendering blobs.
UI\Pings\Pings.lua	ATI templates for updating and rendering pings.

Following sections describe Blobs.lua and Pings.lua in greater detail. Here is the meaning of the various tunables in SensorsManager.lua:

Tunable	Type/range	Description
smTwkBlinkTime	float 0.05 < 2 seconds	How fast selected ships in the SM blink. Dots, meshes and TOs will all blink.
smTwkZoomTime	float 0.05 < 2 seconds	Time it takes to zoom to or from the SM.
smTwkZoomOutDistance	float	Initial SM zoom out distance
smTwkMinZoomOutDistance smTwkMaxZoomOutDistance	float	Sensors manager camera zoom range. Can be overridden on a per-level basis.
smTwkNearClipPlane smTwkFarClipPlane	float	Clip plane range in the SM. The far clip plane should generally be $\geq 2 \times$ max zoom distance.
smTwkMaxMovementDistance	float	How far can we move in the SM.
friendlyPixelColour alliedPixelColour enemyPixelColour resourcePatchPixelColour miscPixelColour	vector4 {0,0,0,0} < {1,1,1,1}	Colour of dots rendered to represent ships, resources and other objects in the SM.
friendlyPixelSize alliedPixelSize enemyPixelSize	float 1<	Size of dots rendered to represent ships, resources and other objects in the SM.

HOMEWORLD2

Copyright © 2003 Relic Entertainment, Inc

Page 1

resourcePatchPixelSize		
miscPixelSize		

### **Sensors Manager Rendering**

In general, the sensors manager is rendered just like the main game view. Following is a list of differences:

- Sensors manager has letterbox bars for now. This will later be replaced with special Sensors Manager UI.
- Most ships, resources, and world objects are rendered as dots.
- Selection graphics and other ATI items have AVF\_MainGameView and AVF\_SensorsManager visibility flags to control differences in ATI between the main game view and the SM.
- Selected ships flash in the SM. This is controlled with the AVF\_SMBlinkOn visibility flag.
- Blobs are only rendered in the SM.
- Most Pings are only visible in the SM.
- World plane graphics only visible in the SM.
- Sensors distortion only rendered in the SM.

### **World plane graphics**

World plane graphics is a special ATI template for rendering ever-present world plane graphics in the Sensors Manager. Currently this consists of a transparent world plane, world plane wireframe graphics and a horizon line. There are visibility flags to control visibility when the camera is above the world plane versus the camera being below the world plane.

### **3D ATI elements**

World plane graphics, Blob and Pings are all ATI templates that can have 3D elements in them. In fact, blobs and pings can have 2D and 3D elements. There are a number of differences between 2D and 3D elements.

### **placement3D**

If an ATIItem has a placement3D instead of a placement2D, the item is 3D. placement3D can have them following members:

Member	Type/range	Default	Description
position	vector3	{0,0,0}	Absolute or relative position of the ATI item.
scale	vector3	{1,1,1}	Absolute or relative per-axis scale of the ATI item.
minArea maxArea	float 0 < 1	0, infinite	Valid range of sizes of the ATI template at which the ATI item will be displayed. Size is in normalized screen size, just like for 2D placements.
visibility visibility0	integer:	0	Strings must be defined in the code to correspond to bit masks. Multiple bit masks will be or'd together to get a final mas. This number will be ANDed with

visibility1 visibility2 visibility3			the blob's ATI visibility mask and must equal itself. {} will always be visible.  There are 4 different possible visibility masks. visibility is an alias for visibility0.
invisibility invisibility0 invisibility1 invisibility2 invisibility3	integer	0	If the visibility test with the corresponding number described above passed, will be ANDed with the ATI visibility mask and if the result is nonzero, item will not be drawn. Useful for excluding visibility in special cases. Like visibility, there are 4 optional masks. invisibility is an alias for invisibility0.
placementFlags	Table of or more of: "absolute"		If true, position and scale will be in absolute world coordinates. If false, position will be added to the ATI template's world position and the scale will be multiplied by the ATI template's scale.
	"scaledPos"		Position will be scaled by the scale parameter.
	spriteXAxis spriteYAxis spriteZAxis		Item will be rendered as a sprite such that it's Z axis faces the camera as best it can, while being free to rotate around one cardinal axis.
	sprite3Axis		Item will be rendered as a sprite such that it's Z axis faces the camera.

## mesh

A 3D ATI item can currently only be a mesh. Meshes can be authored at full size and given a default {1,1,1} scale or they can be modeled with a radius of 1 and given an actual scale. Like 2D mesh items, they should be exported using either the **particle** or **wireframe** export options.

A 3D mesh ATI Item can have the following members:

Member	Type/range	Default	Description
colour	vector4 {0,0,0,0} < {1,1,1,1}	{0,0,0,0}	The colour the mesh will be rendered in. Not needed if a colour parameter is set. If left default, the colour will come from Maya.
lineWeight	float: 1<	1	Thickness, in pixels, wireframe meshes will be rendered at. Only needed for wireframe meshes.
LODs	Table of pairs of LOD value and mesh name.	none	Describes the LOD progression to use. Ordered from highest to lowest detail meshes. LOD value is based on squadron rectangle's size in normalized screen size.

## HOMEWORLD2

## Parameters

Parameters for 3D mesh items are similar to 2D mesh items. The following table lists how the various parameter types are interpreted:

Type	mesh
floatParam	N/A
colourParam	Overrides mesh colour and item colour
stringParam	N/A
positionParam	if "absolute", this will be the absolute world position of the item. If not "absolute", this will be added to the template position.
scaleParam	If "absolute", this will be the x/y/z scale of the item. If not "absolute", multiplied by the item scale.
graphicParam	Specifies mesh item to render

## Blobs

Blobs are rendered as an ATI template. Blob templates have tunables beyond the conventional placement, mesh and parameters:

Member	Type/range	Default	Description
protrude	Float: $0 < 1$	1	If overlapping blobs protrude by this number multiplied by the smaller blob's radius, they will be separate blobs. Otherwise, they will be merged together.
borderPadding	Float $0 <$	0	When creating MetaSelTargs, add this value to the actual blob radius for the selection sphere size.
clickable	bool	0	Blob can be clicked on to select, focus or issue context-sensitive commands.
bandboxable	bool	0	Blob can be bandboxed to select, focus, or issue context-sensitive commands.
metaSelPriority	Integer: $0 \leq 3$	1	Specifies the selection priority of the MetaSelTarg that will be created for the blob. If 0 (NoPriority), no MetaSelTarg will be created.
colourNear colourFar	vector4: $\{0,0,0,0\}$ $< \{1,1,1,1\}$	none	Colour range for the blob sphere. Will be passed back as BPI_BlobColour
colourMouseOverNear colourMouseOverFar	vector4: $\{0,0,0,0\}$ $< \{1,1,1,1\}$	none	Colour range of the mouse when moused over. Passed back as BPI_BlobColour.
colourFootNear	vector4: $\{0,0,0,0\}$	none	Colour range of the foot. Passed

## HOMEWORLD2

colourFootFar	< {1,1,1,1}		back as BPI_FootColour.
colourMouseOverFootNear colourMouseOverFootFar	vector4: {0,0,0,0} < {1,1,1,1}	none	Colour range of the foot when the foot is moused over. Passed back as BPI_FootColour.
colourEquatorNear colourEquatorFar	vector4: {0,0,0,0} < {1,1,1,1}	none	Colour range of the equator, which is the circle that is rendered where the blob intersects with the world plane. Passed back as BPI_EquatorColour.
blobColourCurve	float: 0.2 < 5	1	Exponent of the blob distance colour attenuation curve. 1 is linear.
blobInsideRadius	float: 0.5 < 2	1	Factor of the blob radius where the camera is considered inside the blob.
RUFormatString	String	""	Only used for resource blobs. Used for sprintf'ing the resource count. Formatted string passed back as BPI_RUCount.

Blobs are rendered in 3 passes:

1. 3D items rendered before world graphics (such as ships). They're sorted rendered without z-buffering enabled so they will be behind world graphics. This is needed to get the blobs to appear behind the world graphics.
2. 3D items rendered after world graphics. Z-buffer is enabled.
3. 2D items rendered after world graphics. Like other 2D items, there will be no z-buffering with the world graphics.

Use the BVF\_Background and BVF\_Foreground to control when your 3D ATI items are displayed.

### **Pings**

Pings are the expanding circles in the SM that indicate the location of events. Some pings are visible in the game and some are only visible in the game.

Like blobs, ping display is defined by an ATI template that can contain 2D and 3D elements. See UI\Pings\Pings.lua for said templates.

Pings.lua contains a table called **templates** that contains all the available. Templates are named and this name is used to bind pings to names. For example, Scar scripts refer to these templates by name.

In addition to the standard ATI template data, Ping templates can contain a number of members to define their behavior:

Member	Type/range	Default	Description
minGrowSize maxGrowSize	float	0	The world-space size range of the ping's expanding sphere. If max < min, ping will shrink rather than expand.
minWorldSize	float	0	Minimum size in world space.

## HOMEWORLD2

growTime	float, seconds	0	Time it takes ping to expand from minSize to maxSize
pauseTime	float, seconds	0	Time between ping expansions.
lifespan	float, seconds	0	If nonzero, the duration of the ping. Some pings are meant to last forever and be explicitly deleted.
footColour	vector4	{0,0,0,0}	Colour of foot if not moused over. Passed back in PPI_FootColour.
mouseOverFootColour	vector4	{0,0,0,0}	Colour of foot if moused over foot. Passed back in PPI_FootColour.
metaSelPriority	integer: 0<=3	1	Specifies the selection priority of the MetaSelTarg that will be created for the ping in the main game view, if the ping is visible in the main game view. If 0 (NoPriority), no MetaSelTarg will be created.
metaSelPrioritySM	integer: 0<=3	0	Specifies the selection priority of the MetaSelTarg that will be created for the ping in the SensorsManager. If 0 (NoPriority), no MetaSelTarg will be created.
battleStatusGood battleStatusBad	float: 1< float: <1		Only needed for battle pings. Specifies the friendly/enemy hitpoint ratio needed for PVF_BattleGood and PVF_BattleBad visibility bits to be set.
minWidth minHeight maxWidth maxHeight	float: 0 < 1	0/0/ infinite/ infinite	Lower an upper limits of the on-screen size of the ping's bounding rectangle for the purposes of selection and ATI rendering.
pingSfxPatch	filename		Sound patch to play when ping "pings".
battleGoodPatch/ battleGoodPatch	filename		Sound patch to play when battle is going poorly/well.
emitterSfxPatch	filename		Ambient sound of ping.

As most pings are rendered as an expanding circle, the game computes the screen size of this circle and passes it to the template rendering in the scaleParam PPI\_PingSize.

### Scar functions for creating pings

The following functions are available from SP scripting for creating pings:

```

Ping_AddPoint( <pingName>, <templateName>, <volumeName>)
Ping_AddSobGroup( <pingName>, <templateName>, <groupName>)
Ping_Remove( <pingID> )
Ping_LabelVisible(<pingID>, <visible>)
Ping_AddDescription( <pingID>, <lineNum>, <description>)
```

---

### HOMEWORLD2

Where:

Parameter	Description	Example
pingName	A localizable string identifier. If the ping label is made visible with Ping_LabelVisible, this string will be available through the stringParam PPI_Label.	"\$41900"
templateName	Name of the ATI template to use for the specified ping.	"anomaly"
volumeName	Name of the volume, defined in Maya, about which to centre the ping.	'pnt_Waypoint_2'
groupName	Name of the sob group, defined in Maya, about which to centre the ping. Ping will move and size to fit the specified group.	'inhibitor_1'
pingID	The value returned from Ping_AddPoint or Ping_AddSobGroup	inhibitorPing
lineNum	Line number of description to add; 0..3.	1
description	Description text for particular line.	'\$41989'