

YarHar 2D Map Editor
JSON specification for maps

November 2012

Prepared by: Stephen Lindberg

Contents

1. Overview	3
2. Scope	3
3. Conventions Used	3
4. JSON specification	4
4.1 Top Level	4
4.2 SpriteLibrary JSON.....	5
4.3 SpriteTypeGroup JSON.....	6
4.4 SpriteType JSON	7
4.5 Layer JSON.....	8
4.6 SpriteInstance JSON	9

1. Overview

YarHar is an open-source Java-based UI project available at <https://github.com/Cazra/YarHar>.

It allows the user to create layered maps of 2D sprites, typically for use in computer games. These maps are saved as JSON text so that they can be loaded in almost any programming language.

This specification is for the JSON format used by the YarHar 2D map editor for representing maps manipulated by it. After reading this document, you will have the knowledge for how to convert a map's JSON text into an instance of the map, assuming that you already know how to use JSON in your preferred programming language.

If you are unfamiliar with JSON, go here: <http://www.json.org/>. It's actually very easy to use, and it's also very light-weight.

2. Scope

This specification is used for describing the format used by JSON text produced by YarHar for representing maps.

3. Conventions Used

I use some formatting conventions for the sake of convenience in describing the JSON format in this document. They are as follows:

- Fields with **blue text** do not need to be processed when constructing the map outside of Yarhar. They are only used for editing purposes within YarHar.
- Fields with **green text** are optional, but could be useful in many cases.

4. JSON specification

This section describes the formats for all the JSON components of a YarHar map.

4.1 Top Level

This section describes the JSON format for the topmost layer of a map's JSON text.

```
{
  "yarhar" : {
    "name" : string,
    "bgColor" : int,
    "spriteLib" : SpriteLibrary JSON object,
    "layers" : [list of Layer JSON objects],
    "desc" : string,
    "gridSpaceX" : int,
    "gridSpaceY" : int,
    "gridW" : int,
    "gridH" : int,
    "gridColor" : int
  }
}
```

"yarhar" : Every YarHar map has this object at the very top of their JSON hierarchy.

"name" : The file name of this map.

"bgColor" : the RGB integer value for the map's background color.

"spriteLib" : The JSON object representing the sprite library used to render sprites used in the map in YarHar.

"layers" : Our map's list of layers, ordered from top to bottom in rendering order.

"desc" : A description of this map for documentation purposes.

"gridSpaceX" : The current horizontal spacing of vertical grid lines in YarHar for this map.

"gridSpaceY" : The current vertical spacing of horizontal grid lines in YarHar for this map.

“gridW” : The width of the grid in YarHar for this map.

“gridH” : The height of the grid in YarHar for this map.

“gridColor” : The RGB integer value for the grid in YarHar for this map.

4.2 SpriteLibrary JSON

This section describes the JSON format for the SpriteLibrary object. The SpriteLibrary serves two primary purposes in YarHar:

- Allow the user to drag SpriteTypes from the library into the editor area to create new SpriteInstances in the current layer.
- Provide a mapping of type strings used by SpriteInstances to their corresponding SpriteTypes for rendering the instances properly in the YarHar editor area.

The SpriteLibrary is really only used by YarHar and it is not needed to create a map instance outside of YarHar. I document its format here anyways in case someone does have an outside use for it.

```
{  
    "groups" : [list of SpriteTypeGroup JSON objects],  
    "sprites" : [list of SpriteType JSON Objects]  
}
```

“groups” : List of sprite type groups used in the library’s interface for organizational purposes.

“sprites” : List of the sprite types used by the map’s library.

4.3 SpriteTypeGroup JSON

This section describes the JSON format for SpriteTypeGroup objects. The only purpose of a SpriteTypeGroup is to categorize SpriteTypes in a SpriteLibrary, purely for user convenience in YarHar.

Each SpriteTypeGroup in a SpriteLibrary contains a set of SpriteTypes that is a subset of the SpriteTypes contained in the SpriteLibrary. The sets of SpriteTypes in a SpriteLibrary's SpriteTypeGroups also form a partition of the SpriteLibrary's set of SpriteTypes. That is, every SpriteType in a SpriteLibrary is contained in exactly one SpriteTypeGroup in that SpriteLibrary.

Like the SpriteLibrary, this is really only used by YarHar and it is not needed to create a map instance outside of YarHar. I document its format here anyways in case someone does have an outside use for it.

```
{  
    "name" : string,  
    "sprites" : [list of SpriteType JSON Objects]  
}
```

"name" : The name of this group of SpriteTypes.

"sprites" : The set of SpriteTypes contained in this group.

4.4 SpriteType JSON

This section describes the JSON format for SpriteType objects.

SpriteTypes are used in YarHar for defining how a SpriteInstances matching the type is displayed in the editor area and in the SpriteLibrary's interface.

Like the SpriteLibrary, this is really only used by YarHar and it is not needed to create a map instance outside of YarHar. I document its format here anyways in case someone does have an outside use for it.

```
{
    "name" : string,
    "img" : string,
    "cx" : int,
    "cy" : int,
    "cw" : int,
    "ch" : int,
    "fx" : int,
    "fy" : int,
    "tc" : int
}
```

"name" : The ID of this SpriteType used to catalog it in a SpriteLibrary.

"img" : The path to the image file used to display SpriteInstances associated with this SpriteType.

"cx" : The x coordinate of the left edge used to crop the image.

"cy" : The y coordinate of the top edge used to crop the image.

"cw" : The width of the image's cropped area. -1 if cropping is not used.

"ch" : The height of the image's cropped area. -1 if cropping is not used.

"fx" : The x coordinate of the sprite's focal point. (The relative position of the sprite's actual position to the upper-left corner of its image)

"fy" : The y coordinate of the sprite's focal point.

"tc" : The RGB integer value of the transparent color used in the image.

4.5 Layer JSON

This section describes the JSON format for Layer objects.

Layers serve the purpose similar to how layers are used in popular graphics editing programs like Flash, Paintshop, or Photoshop. They are used to divide sprites on a map into disjoint groupings above/below each other.

This allows sprites to be placed, for example, in a background layer appearing behind everything else, in a middle layer where all the interactive bits of a map are placed, or in the foreground layer appearing above everything else.

```
{
    "name" : string,
    "sprites" : [list of SpriteInstance JSON objects],
    "opacity" : float,
    "visible" : boolean
}
```

"name" : The name of this layer displayed in the YarHar layers interface.

"sprites" : The list of SpriteInstance in this layer in rendering order from bottom to top.

"opacity" : A floating point value in the range [0.0, 1.0] used to set how opaque the layer appears in the YarHar editor area. 0.0 makes the layer completely transparent. 1.0 makes the layer completely opaque.

"visible" : If true, this layer will be visible in YarHar's editor area. If false, it will be invisible and non-interactive.

4.6 SpriteInstance JSON

This section describes the JSON format for SpriteInstance objects.

SpriteInstances are instances of SpriteTypes placed in the Layers of a YarHar map. Ultimately, these objects will be used to determine where to place the actual sprites in for the map in your game.

```
{
    "type" : string,
    "x" : float,
    "y" : float,
    "a" : float,
    "o" : float,
    "s" : float,
    "sx" : float,
    "sy" : float
}
```

"type" : This is the ID of the SpriteType associated with this instance. Use this in your game to determine what type of sprite to make this instance as using your own sprite classes.

"x" : The instance's x position.

"y" : The instance's y position.

"a" : The angle of this instance's rotation counterclockwise from the positive x-axis, in degrees.

"o" : This instance's opacity, in the range [0.0, 1.0]. 1.0 makes it completely opaque. 0.0 makes it completely transparent.

"s" : The universal scale transform value for the instance. The x and y axis of the sprite's image will both be scaled by this amount. This value is in the range (0.0, +infinity), with 1.0 being its original size.

"sx" : The scale of the instance's image on its x-axis. This value is in the range (0.0, +infinity), with 1.0 being its original size. This stacks multiplicatively with the universal scale transform for the instance.

"sy" : The scale of the instance's image on its y-axis. This value is in the range (0.0, +infinity), with 1.0 being its original size. This stacks multiplicatively with the universal scale transform for the instance.