LDraw Tutorial Part 12

Managing MILS with BlueBrick

by Jetro

MILS
Modular Integrated Landscaping System

In HispaBrick Magazine® 005 I presented BlueBrick, a visual management application for layouts and displays that can be very useful for teamwork on displays for events. Since the collaborative concept of MILS has been presented in this issue I will dedicate this part of the LDraw tutorial to show you how to use BlueBrick to organise collaborative displays with MILS.

Why BlueBrick?

The concept of MILS serves two purposes:

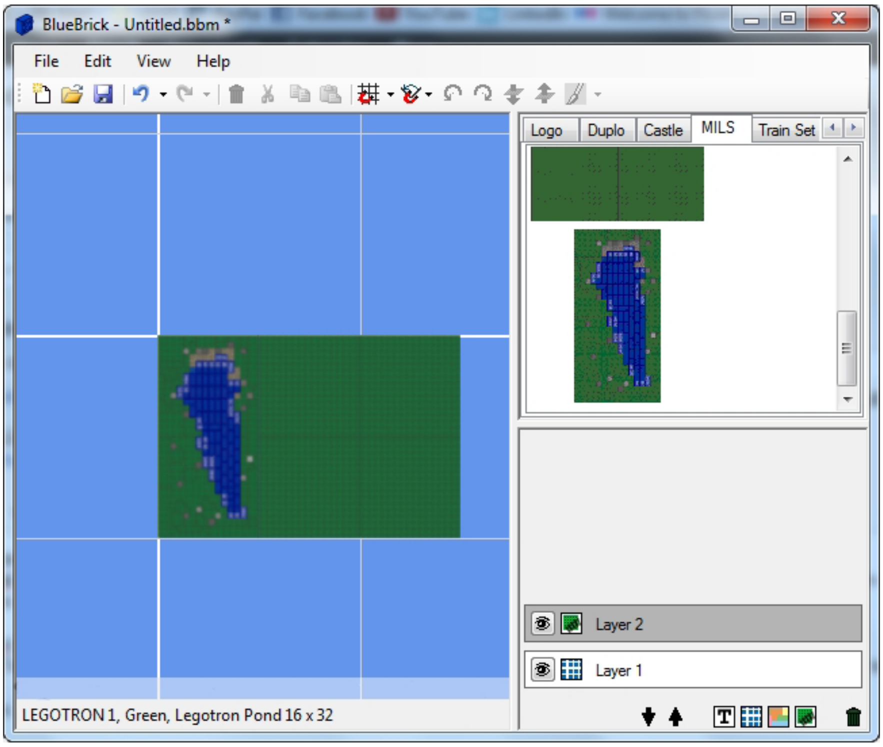
1) it is modular, which means it can be built in different sizes and configurations
2) it makes collaborating easier; otherwise there would be no need for a common standard.

This means that a display created with the aid of MILS can be easily adapted to the available space and allows for the collaboration of several people in a single display. Using BlueBrick, this collaboration can be planned easily, without the need of a physical meeting of everyone involved.

Creating MILS for BlueBrick

Let's start with the simplest module: a flat BTM[1] Since the top view of a BTM is identical to a 32x32 baseplate we can use the existing image from BlueBrick to create our first BTM.

If you don't have the program installed you can download it from the BlueBrick website[2] The program doesn't require any installation as it runs from the same folder you extract it in [3]. To simplify matters, in this tutorial all file locations will be indicated starting from the "BlueBrick" folder.
The elements contained in the BlueBrick parts library are located under \BlueBrick\parts. You will see there are a number of folders inside this folder for each type of element (for example \BlueBrick\parts\Baseplate). These folders determine the names and number of tabs that appear in the BlueBrick parts panel in the top right corner of the BlueBrick window. Since in this tutorial we will be creating a new type of element we should start by creating a new folder called MILS (\BlueBrick\parts\MILS - the folder will show up as a tab the next time you start BlueBrick). From the folder \BlueBrick\parts\Baseplate we will copy the files 3811.2.gif and 3811.2.xml and paste them into the folder \BlueBrick\parts\MILS you created previously. Since the file will be used for a Basic Terrain Module, we will rename them to BTM.2.gif and BTM.2.xml respectively[4].
Next we will modify the XML file that is associated with the image. In order to modify the XML file you need to open it in a text editor like Notepad (in windows you can do so by right clicking on the file and choosing one of the programs from the "open with" entry in the contextual menu that appears). You will notice the file contains the name of the element/module. Between the tags <Description></Description> there are two



LEGOTRON 1, Green, Legotron Pond 16 x 32

pairs of language tags. For now only the tags <en> for English and <fr> for French are available. If you wish to prepare your file for possible future translations you can add a language pair, e.g. <es></es> for Spanish. The text between these tags is the name that is shown in the bottom bar of BlueBrick when you place the cursor over the miniature of the element in the MILS tab of BlueBrick. Change the text between both pairs of tags to BTM.
The information that appears further down in the XML file is specific to remapping the file for Track Designer (TD). Since TD does not have and BTUs, but the shape and colour of this one is identical to a 32x32 baseplate we can leave the information as is for better compatibility.

Creating a personalised module

The next step is to create a representation of one of your own modules. Among the images shown in the MILS article there is one with a pond. How can we make this module available in BlueBrick? There are different ways of doing this and for this tutorial I will describe only one that has given me good results.
The first step is to recreate a top view of the module in an LDraw editor (MLCad, LeoCAD,LDCad, Bricksmith...). For this tutorial I recreated the module in MLCad. Since we need a top view with minimal distortion and neutral lighting, a simple method of obtaining one is to make a screen capture of the top view directly from MLCad. To this end, maximize the MLCad window and make the top view take up the biggest area you can. Next open the context menu (right click) and select the zoom level “fit” to see the module as big as possible in this windows size.
Pressing the “Prnt Scr” button you take a screen capture from which you will have to cut out the module. To this end I have used the free image editor GIMP which has a GNU license. Open a new file in GIMP with a size that is equal or bigger than your screen resolution. Now paste the screen capture you made into the file (Ctrl+V or Edit > Paste). Next choose the rectangle select tool (top left in the left floating panel) and roughly select the module, making sure there is only white around it. Under the menu “Image” you will find three very useful tools we will use now. The first one is “Crop to selection” which will leave only the area you selected previously. Now that you only have white borders around the image you can use the option “Autocrop Image” which will automatically eliminate this white border. Finally you will need to adjust the size of the image to the size BlueBrick needs: the image should be 8 pixels for each stud in width or length. In the case of a 32x32 baseplate the image size should be (8x32) 256x256. You can do this with the option “Scale image...”. Now you need to change the image to GIF format. You can do this from File > Save as and selecting the file type you need.
What’s left is creating the corresponding XML file. Just like we did for the BTM above, it is convenient to look for an existing element that is as similar in size as possible and use a copy for the module we have just created. After that you need to follow the same steps as for the BTM described above. In order to make it easier to work as a group it would be convenient to use a module name that identifies its author as a first element of that name. Since BlueBrick shows elements in alphabetical order, this way you make sure that all modules from the same author are grouped together.
This is as far as the MILS article went and so this is where I finish this part of the LDraw tutorial. In the next issue I will continue explaining how to adapt what is explained in the MILS article to managing modules with BlueBrick.

[1] The following example is a simple way to create a BTM from an existing element. You can download a BTM specifically created for the MILS group (image and XML file) from the HispaBrick Magazine® website.

[2] http://bluebrick.lswproject.com/

[3] The most recent version of BlueBrick at the time of writing this article is 1.7.1. This version includes some changes with respect to earlier versions to make it fully compatible with Mono and so be able to use the program under Linux as well as Windows. If you have an earlier version it is highly recommended to update to the latest version to make sure all elements are fully compatible.

[4] BlueBrick extracts information about an element from different places. The colour of the element is indicated in the name of the file. A 32x32 baseplate (BL code 3811) has an element name plus a suffix to indicate its colour (both in the GIF and the XML). In this way a blue 32x32 baseplate would be named 3811.1, a green one 3811.2 and a tan one 3811.19. The colour code corresponds to the LDraw colour numbering. To distinguish a green BTM (for a field) from a white one (for snow) this suffix can also be used in the names for these elements.
#