Patching Basics

(A very good overview is also contained in the v9 help, to be found in Distrib\Help\Ai_Man_v8.chm and in the V9 User Manual available at the Avolites Download site.).

In the beginning - when it was just the Salvation engine - almost everything needed to be patched: the engine offered a collection of modules, each specialising in a very small task, and the operator/programmer had to combine them in a meaningful way, to acchieve what he had in mind - play a video, map an image to a structure, control some parameters. A good part of the development over the last years is that many of everydays tasks are now packed into a fancy GUI, and are available with just one click. However, if something special needs to be accomplished, or some particular thing is not (yet) available via the GUI, patching might be the way to go. And honestly, after you have made the first steps, it is pure fun (after the desperation has vanished, that is).

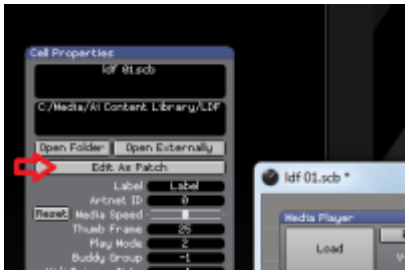
Where to start

Currently there are some entry points into the world of patching:

- the purest approach is the File/New menu in the Ai menu bar on top of the GUI display - this opens just a blank patch file
- the mightiest way is clicking Stage Patch: `Open` button in the little Ai root window. This opens a patch window which is actually the current project, with some modules and subpatches. Personally I prefer this as entry point - but you must know that you can easily render your system completely useless here.
- a more fancy way is calling the Stage Patch page from the Ai diamond in the bottom-right corner - click this, then click the stylized single DVI connector (outputs), and then the stylized wired modules (Stage Patch). This again opens the Stage Patch but in a more stylish view: you can zoom in and out (with Ctrl + and Ctrl -), entering a subpatch opens this neatly animated, and in order to ascend again you have the breadcrumps on top. It's up to you which way you prefer.
- and a special one: even all the tiles on the performance page are, down in their heart, small patches. Hence, if you right-click a tile to edit its properties the Cell Properties window to the left shows the button `Edit As Patch`. Clicking this opens a patch window with the modules assigned to this particular cell.

To start patching simply open a blank patch file, or open the Stage Patch.





Patches, Ports and Modules

In the very end, the stage patch consists of just thousands of modules, all linked to other modules. However none could overlook this or handle it in a suitable way. That's why functional units are bundled together as subpatches which expose only a limited set of ports and parameters. By convention every subpatch, or patch file, is simply called patch, with the Stage Patch representing our topmost project, containing all other patches and modules (it's possible to have patches outside the Stage Patch though). And the process of patching is arranging and connecting modules.

Modules



Modules are the smallest parts of Ai. They are provided by the Salvation engine which is the very core of Ai. Hence, you cannot create modules yourself, but you can combine the existing modules in a meaningful way. For a full list of the available modules please see Distrib\Help\Ai_Man_v8.chm or the Module Definitions section in the V9 User Manual which is available at the Avolites Download site.

In order to insert a module right-click somewhere on the background of your patch canvas and from the context menu select `Insert Module`. This opens a list of sections which the available modules are grouped into. When you hover the mouse over a section, another list appears showing all the modules in this list. Simply click a module to select it, then place your mouse where you want the module to be inserted, and click again. This places the chosen module at this spot.

There is a separate page about the details of re-arranging, moving, customising and resizing modules/patches and their controls: Customizing Patches and Modules.

Modules are represented by grey rectangles with a blue title bar and a number of controls (buttons, text fields, knobs etc.) and ports.

Ports

Ports are the inputs and outputs of the modules, and are shown as triangles. If they are pointing up then they are the module's inputs, down-pointing ports are outputs. There are several colours used (in more detail explained in the manual):

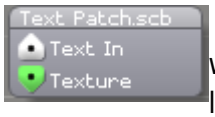
- blue ports are control values - usually single numbers, or vectors comprising of a bunch of numbers (e.g.RGBW)
- green ports are textures - e.g. 2D images and videos
- yellow ports conduct the rendered video stream

- white ports are for text and other multidimensional things

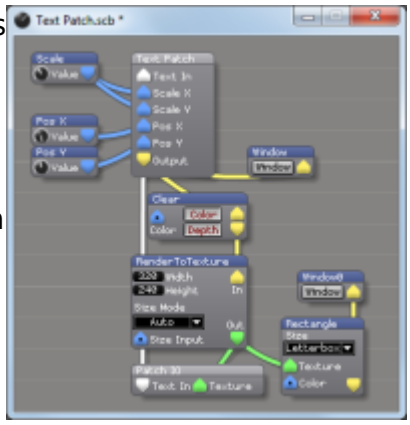
Connecting ports is rather easy: just draw a line from one output port to an input port (or vice versa) - as soon as you drop the other end of the virtual cable onto a suitable port, a line appears, indicating the connection. There are some rules to follow:

- ports can be connected only with other ports of the same colour/kind
- each input accepts only one source, however outputs can feed multiple inputs
- ports which are currently not connected appear with a dot in the middle
- in order to disconnect a port/remove a cable, right-click the port and select `Disconnect`

Patches



As mentioned above, patches are collections of modules where only limited ports and controls are exposed. They look similar to modules with the difference that the title bar is grey. However, in order to look what's inside this patch, you can just couble-click it, and the patch opens in another window (or in case you are using the Stage Patch page, you smoothly fly into this patch). In the case of the shown (left picture) text patch (which is available as system patch - right-click, `Insert System Patch`, select 'Text Patch.scb'), the contents is this (right picture):



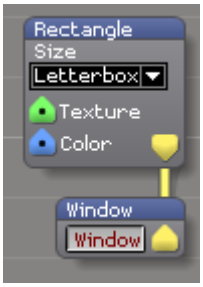
There is a separate page about the details of re-arranging, moving, customising and resizing modules/patches and their controls: Customizing Patches and Modules.

Our first patch

You're still with me? Let's do it. Here is your first patch:

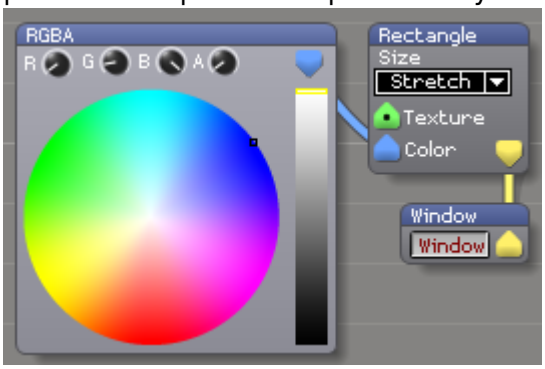
1. open a new patch file as decribed above (preferably select File/New from the top menubar)
2. in the blank patch window, right-click somewhere, select 'Insert Module', and from the section 'GL' select the 'Window' module. Click again somewhere in the patch window to place the module there - it will look like this:
   

   If you click the button labelled `Window` then a new window will open, showing a black canvas.

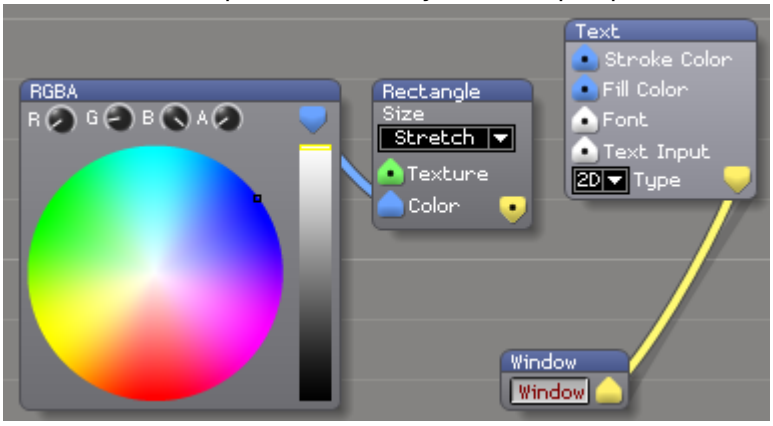
3. Next, we want to make the background other than black. Insert a 'Rectangle' module from the 'Generator' section, and wire its yellow output port into the window module's input port:
   

the output window now shows a white rectangle in the middle. You might want to change the only parameter in the 'Rectangle' module from 'Letterbox' to 'Stretch', to make the whole window white.

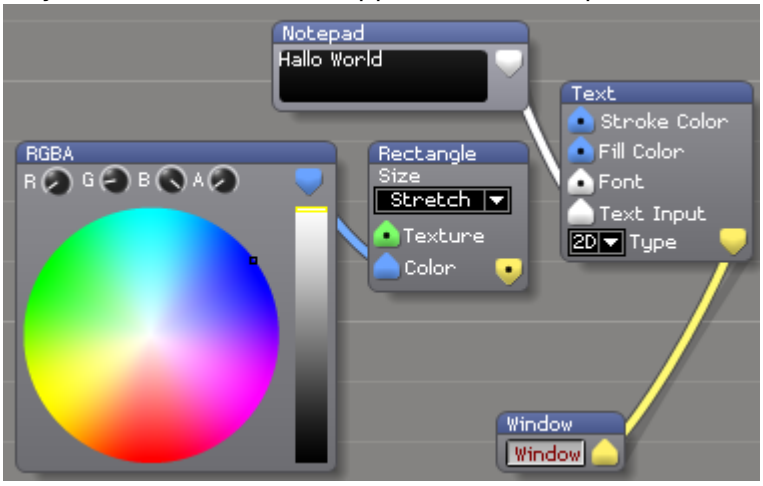
4. Okay, let's bring in some colour. Insert an 'RGBA' module (section 'Colour'), wire its blue output port to the blue input port labelled 'Color' of the Rectangle module, and play with the colour picker or the parameter pots - now you can adjust the colour of the output:



5. Now we want to show some text. We want to generate something for the output, hence let's look in the Generator section. We'll find a 'Text' module which looks rather promising. Lets insert it into our patch. It has a yellow output port which we wire to our window module.

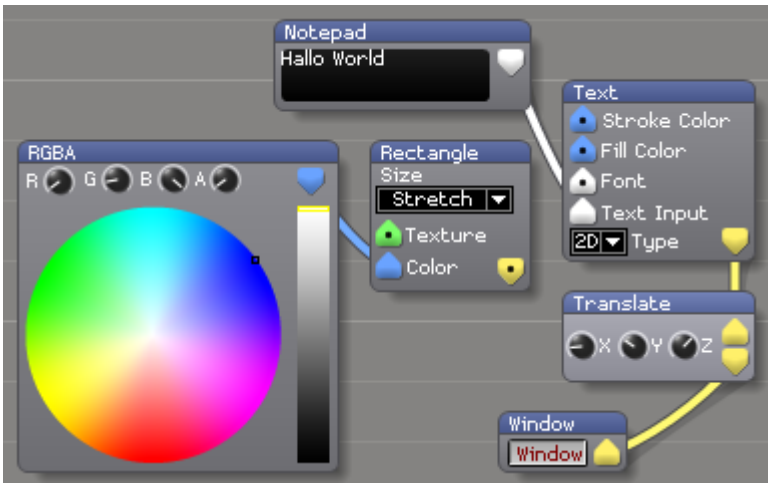


6. But as soon as we connect the yellow port, the output window turns black - because each input can take only one output (thus the connection to the coloured rectangle went lost), and because the Text module has no text input to display. Let's proceed with this. From the 'Text' section, insert a 'Notepad' module. Wire its white output port into the white 'Text Input' port of the Text module, and finally type something into the text area of the Notepad module. As soon as you hit Enter, the text appears in the output window:



7. However, it looks like the text always starts in the center of the window. If we resize the window, the text resizes and repositions accordingly. In order to fix this, we obviously need to reposition the rendered text output. The module 'Translate' from the 'GL' section does this for us - insert it, and rewire the connetcions so that it sits between the Text module and the

Window module.



Now you can comfortably adjust the text position with the XYZ knobs in the Translate module.

8. In order to finally combine the generated text with our colourful background we need to somehow combine both video streams. The module 'Render Merge' from the GL section does this for us. Insert it, rewire the video stream so that the background goes into input 1, the translated text into input 2, and the output is sent to the window. The result should finally look like this:



Congratulations! You created your first own Ai patch. Was it really that complicated?

From:
https://www.avosupport.de/wiki/ - **AVOSUPPORT**

Permanent link:
https://www.avosupport.de/wiki/ai/patching_basics?rev=1539462722

Last update: **2018/10/13 20:32**