

# apolloensemble

imagine create

## user guide





## Warning

This is a Class A product. In a domestic environment this product may cause radio interference.

This equipment is marked with the recycling symbol on the right. It means that at the end of the life of the equipment you must dispose of it separately at an appropriate collection point. Products bearing this symbol must not be discarded together with household waste.



Although Apollo Creative has tested the Apollo Ensemble with typical computers that meet the system requirements, we cannot guarantee that a given computer can be used based solely on the fact that it meets the criteria. This is due to the fact that there are too many variables that may influence the performance including processor, motherboard design, sound cards and the combination of other devices involved.

### Caution

- Try to prevent cords and cables becoming entangled.
- Do not place heavy objects on the Ensemble16 or EnsembleHub.
- Disconnect all cables before moving the system.

The Apollo Ensemble is designed and manufactured by Apollo Creative. For more information see our website at <http://www.apollocreative.co.uk>.

Apollo Creative  
Knaresborough Technology Park  
Manse Lane  
Knaresborough  
North Yorkshire  
HG5 8LF

Tel: 01423 798258  
Fax: 01423 798239  
Email: [sales@apollocreative.co.uk](mailto:sales@apollocreative.co.uk)

- Microsoft and Windows are registered trademarks of Microsoft Corporation.
- Apple is a registered trademark of Apple, Inc.
- All product names mentioned in this document are trademarks or registered trademarks of their respective owners.

## TABLE OF CONTENTS

<b>Version Information .....</b>	<b>1</b>
<b>1. Introduction .....</b>	<b>2</b>
1.1 <i>The manual</i> .....	3
1.2 <i>What is Apollo Ensemble?</i> .....	3
1.3 <i>How does Apollo Ensemble work?</i> .....	4
1.4 <i>More about movement, sounds, light and video</i> .....	5
<b>2. Ensemble Software .....</b>	<b>9</b>
2.1 <i>Concept</i> .....	9
2.2 <i>The Designer</i> .....	9
2.3 <i>The Player</i> .....	14
<b>3. Creating and running a map in Designer .....</b>	<b>16</b>
3.1 <i>Tutorial Example 1: Triggering a sound from a switch</i> .....	17
3.2 <i>Tutorial Example 2: Changing the volume using a second switch</i> .....	26
3.3 <i>Tutorial Example 3: Using a variable sensor</i> .....	29
3.4 <i>Further work</i> .....	31
<b>4. Blocks Reference .....</b>	<b>32</b>
4.1 <i>About Blocks</i> .....	32
4.2 <i>'Input' Palette</i> .....	34
4.3 <i>'Tools' Palette</i> .....	36
4.4 <i>'Lighting' Palette</i> .....	44
4.5 <i>'Sounds' Palette</i> .....	46
4.6 <i>'Video' Palette</i> .....	59
<b>Appendix 1: Glossary .....</b>	<b>62</b>
<b>Appendix 2: The Menus and Toolbar .....</b>	<b>64</b>
<b>Appendix 3: What is MIDI? .....</b>	<b>67</b>
<b>Appendix 4: What is DMX? .....</b>	<b>67</b>
<b>Appendix 5: Declarations of Conformity .....</b>	<b>68</b>



## Version Information

### Amendment History

Date	Reference	Details
04.08.09		Apollo Ensemble Version 1.0 Manual
25.05.10		Apollo Ensemble Version 2.0 Manual
22.02.11		Updated images – pgs 4, 6; 'Sounds' block icons updated; MIDI Controller Block added.
16.04.12		Added DMX Scene; new toolbar; new screenshots

### Acknowledgements

Original manual written by Tim Anderson.

Additional material by Mark Hildred and Ben Johnson.

Design and formatting by Empathy Design ([www.empathydesign.co.uk](http://www.empathydesign.co.uk)).

## 1. Introduction

---

### **Welcome to the Apollo Ensemble.**

The Apollo Ensemble is the culmination of many years work and combines my experience of working in several different but overlapping areas. It began as a method to allow people with physical disabilities to play music and slowly grew to be used for creating interactive multimedia environments. Along the way we added the ability to control lighting and effects as well as video and images. This flexibility has opened up many areas of application from musical performance and exhibitions to creating multisensory rooms and children's play areas.

At the heart of the Ensemble is the Designer, an easy to use piece of software that allows you to decide how actions in the real world are turned into music, sound, lights and special effects. This drag and drop software allows you to create setups from simple cause and effect, through to fully themed environments.

The Ensemble interface is your link to the real world, used to connect your computer to sensors and switches. These can be selected to detect a range of different movements and gestures – from squeezing to rolling on the floor; waving a hand to walking into a room. You can choose these to suit your particular application, either using existing switches or selecting from Apollo's growing range of wired and wireless sensors.

This manual contains all the information you need to get started, from setting up to creating your own themes. Don't forget to also visit our online support site at <http://support.apolloensemble.co.uk>, where you'll find additional resources to help you get the most from your system.

Good luck – and enjoy using the Apollo Ensemble!



Mark Hildred  
Creative Director, Apollo Creative

## 1.1 The manual

This manual has been written to present you with an introduction to some of the key concepts required for using the Apollo Ensemble software.

Although this guide tries to use simple language, it is impossible to escape the use of some specialised words, such as 'blocks', 'signals', and 'sensors'. These can be confusing at first and it is recommended that you refer to the glossary (Appendix 1) while reading, to help understand these words and their relationships.

## 1.2 What is Apollo Ensemble?

**Apollo Ensemble** is a system that allows you to detect *movement* (of your body or other objects) and convert it into *sound*, *light* and *image*.

- Movement: pressing a switch, touching or squeezing a pad or pressure sensor, or even waving freely in space with a position sensor.
- Sound: playing musical notes in a scale, a set of chords, sound effect clips, musical loops or samples.
- Light: flashing a lamp on/off, changing brightness or colour of a light, or moving a motorised spotlight.
- Image: triggering still or moving images on a screen or video projector.



A portable Apollo Ensemble set-up with laptop, EnsembleHub, switch and EnsembleDice sensor.

**Ensemble** is very flexible, for example:

- You can trigger a sound or turn on a light using a single switch or sensor, control many things at once or make them happen in order.
- You can control things in a more sophisticated way, linking switches and various sensors to many different effects.
- You can link Ensemble to specialist music software, such as 'Ableton Live' or 'E-Scape', to access advanced features.
- You can use Ensemble with many people at the same time, encouraging them to interact with each other – for example one person could trigger a sound and a light, whilst someone else's position could affect the colour or volume, or even both at once.

The Apollo Ensemble has a wide range of possible uses, from music concerts, music therapy sessions, music or dance workshops to interactive installations, multi-sensory rooms and dramatic presentations.

### 1.3 How does Apollo Ensemble work?

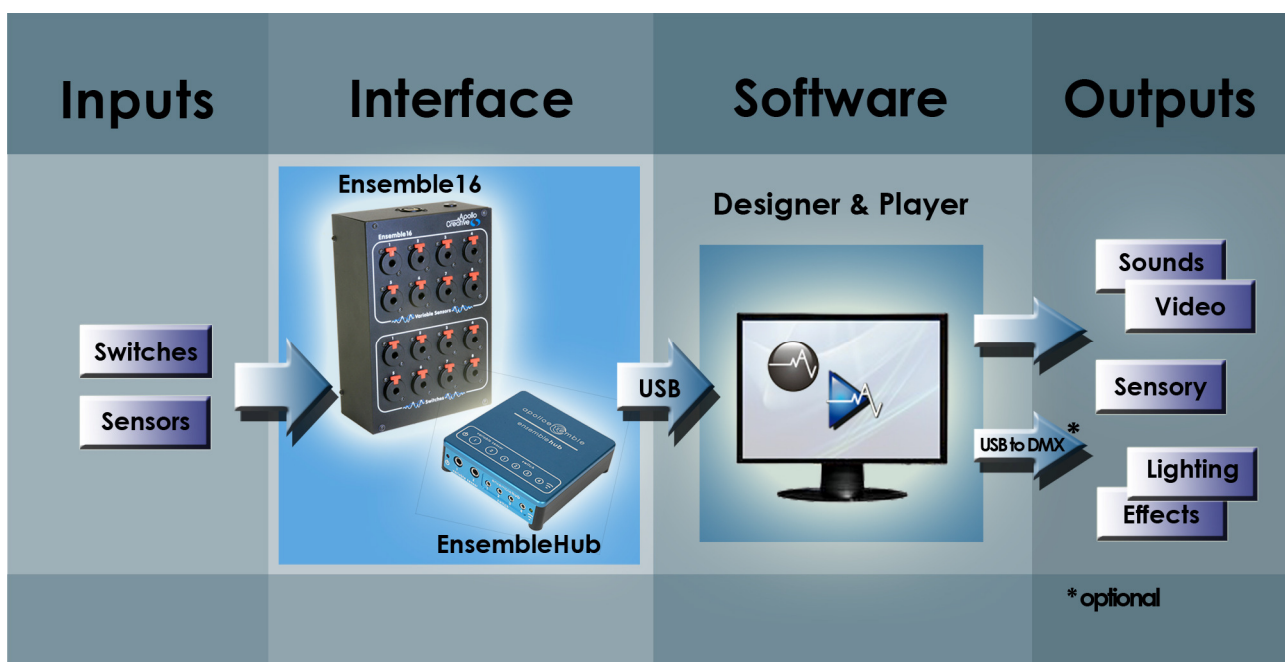
Apollo Ensemble has two main parts – the interface and the software.

The interface connects the PC to the real world and has sockets into which you plug your switches or sensors. This connects to a Windows PC via a USB cable.

The Ensemble software runs on the PC, and lets you setup the system and converts messages from the interface into sound and light etc. The software has two parts:

- 'Designer' lets you decide how inputs are converted into sound and light, "what will happen".
- 'Player' runs to actually operate the system when you are performing.

You can use the Player on its own to run set-ups created by other people, *without* needing to know about Designer.



## 1.4 More about movement, sounds, light and video

### 1.4.1 Detecting Movement

You can connect switches and sensors to the Apollo Ensemble interface to detect a wide variety of movements made by people, or objects.



A 'switch' is a simple sensor just like a light switch, which can be in two positions: 'on' (pressed) or 'off' (not pressed). Either position can make something happen.



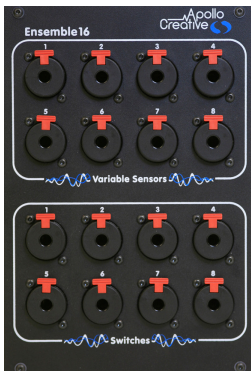
A 'variable sensor' is more like a dimmer or volume control, and has a *range* of values or positions. It can change things like brightness or volume in a simple way, or can be converted into more complicated sound or light effects.

You can use any assistive technology switches (available from specialist suppliers), as switch inputs. These cover a range of actions including tilting your head, sipping and blowing and tapping. It is also possible to make your own from commonly available components.

Apollo Creative supplies a number of wired and wireless sensors. Again it is possible to create your own variable sensors and connect sensors designed for other systems. See the support website for further details.

The switches and variable sensors connect to the PC via one of the two Ensemble interfaces.

- Ensemble16 – is housed in a black metal stage box and is designed for fixed, wired installations.
- EnsembleHub – is a compact, portable interface which as well as having wired inputs acts as a receiver for wireless switches and sensors.



## Ensemble16

Up to eight variable sensors can be connected via the top eight sockets, plus eight switches connected at the same time via the bottom eight sockets. The sockets take standard 1/4" (6.35mm) jack plugs, which *lock* into the sockets - press the red tab and pull the plug to release them.

If your switches or sensors come fitted with the smaller 3.5mm jack, then a common 'headphone adaptor' can be used to plug them into the 1/4" sockets.

Make sure you plug switches or sensors into the correct type of socket:



A **sensor** has a 'stereo' jack plug (with three wires - 'ring', 'tip' and 'sleeve'), and connects into one of the top eight sockets.

A **switch** has a 'mono' jack plug (with two wires - 'tip' and 'sleeve'), and connects into one of the bottom eight sockets.



## EnsembleHub

Up to two variable sensors can be connected directly to the Hub via the 1/4" jack sockets, plus four switches via the 3.5mm sockets. The hub also links to wireless switch interfaces and sensors.

Make sure you plug switches or sensors into the correct type of socket:



A **sensor** has a 'stereo' jack plug (with three wires - 'ring', 'tip' and 'sleeve'), and connects into one of the four sockets labelled 'variable sensor'.

A **switch** has a 'mono' 3.5mm jack plug (with two wires - 'tip' and 'sleeve'), and plugs into one of the four sockets labelled 'switch'.



In addition the Hub can control Apollo Creative interactive bubble tubes and fibre optic light sources.



### 1.4.2 Sound

Two kinds of sound can be created by the Ensemble:



- Notes or chords can be played using 'MIDI instruments' (see Appendix 3 'What is MIDI?' for more details). These MIDI instruments can be the ones built into Windows; higher quality 'soft synth' or sampler software on your PC; or an external MIDI module or keyboard connected via USB or MIDI cable.
- Audio recordings ('samples') can be played in various ways.

Sounds can be controlled and triggered in a wide variety of ways. To give some simple examples:



- Each press of a switch can trigger a single note, chord or sample.
- A switch or sensor can play a series of notes or chords, arranged in any order.
- Delays can be put between sounds.



- Sounds can start, then can trigger other sounds or lights when they finish.
- Each position within a range of movement can play a different note or sound.



- The volume or attack of sounds can be changed via a range of movement.
- The instruments in use can be changed by the press of a button.

### 1.4.3 Lighting & Effects

Lighting and effects (i.e. theatrical spot-lights or disco lighting) can also be controlled by Ensemble, if they conform to the 'DMX' standard (see Appendix 4 'What is DMX?'). These connect to the PC using a separate USB-to-DMX interface. They can be controlled in many ways – what you can do depends on the capabilities of the particular lights you use, for example:



- simple flashing on and off;
- changing brightness and/or colour, either continuously or to preset values;
- moving the direction of the light beam – e.g. if you have a 'motorised' spotlight (moving head or scanner).

Other devices can also be controlled using a DMX 'dimmer pack' or 'switch pack' interface, which allow you to switch any mains-powered item, such as a fan, on or off.

#### 1.4.4 Image and Video

Images and video can be triggered and controlled by the Apollo Ensemble. They can appear on the computer monitor or on a second screen if the PC has an additional monitor output. You can use images and videos in several ways, for example:



- Each press of a switch triggers a different video.
- Each position within the range of a sensor can trigger a different image.
- Videos can play in a sequence, with or without sound.
- Videos and images can appear in different parts of the screen and in different ways e.g. fading or flying in.

The types of image and video files that can be played back depend on the 'codecs' installed on your computer. A selection of standard video codecs can be installed from the software CD or downloaded from the internet.

---

## 2. Ensemble Software

### 2.1 Concept

The Ensemble software consists of two parts:



- 'Designer' enables you to design and set up what will happen when you are operating.

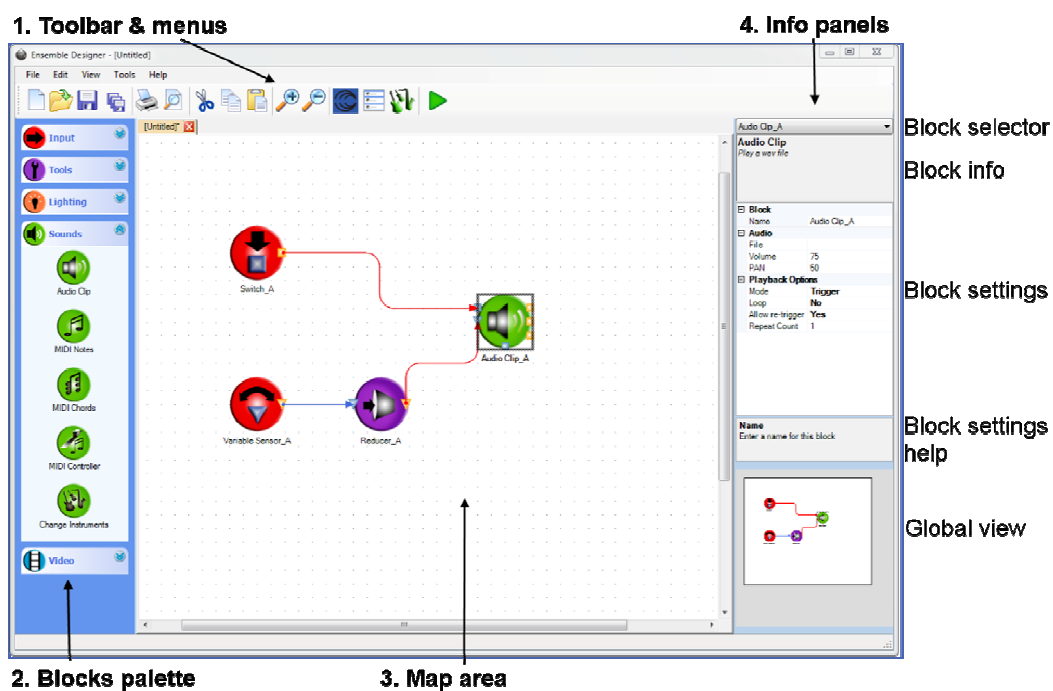


- 'Player' then runs the system when you are 'live' and operating.

### 2.2 The Designer

Let's start by having a look around Ensemble Designer. If it's not open, then open it now - either from the desktop shortcut (see above) or via Windows Start > 'All Programs' menu.

#### 2.2.1 A look at the Designer screen

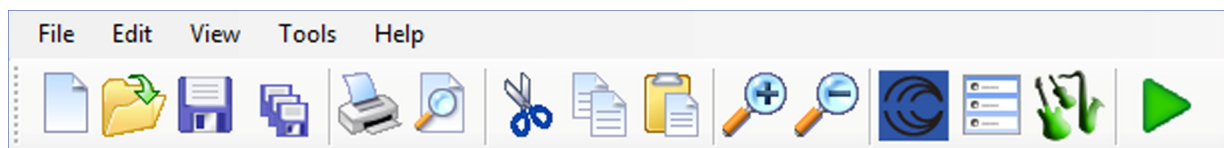


There are four main sections in the Designer window:

## 1. Toolbar and menus

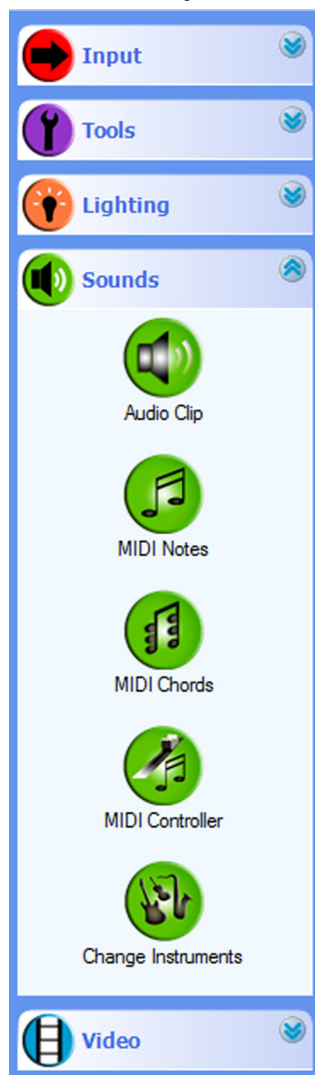
At the top is a toolbar with buttons to do common jobs such as save and load, print, copy and paste, and zoom in or out. All these options are also in the drop down menus. If you move the mouse over a tool, and wait for a short time, it tells you what it is. On the far right of the tool bar is the 'Play' button, which you press to start the map running (this actually opens the 'Ensemble Player' – see section 2.3).

See Appendix 2 for further details.



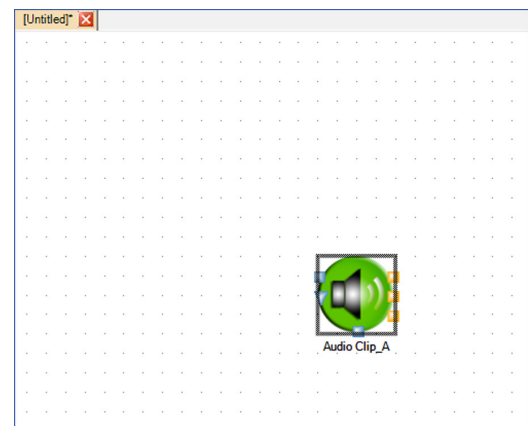
## 3. Map area

## 2. Blocks palette



On the left is a set of 'palettes', which contain 'blocks' to do different jobs. For example, the 'Input' palette has blocks that receive input from switches or sensors, and the 'Sounds' palette has blocks that create and control sound.

You can expand or collapse each palette by clicking anywhere on its title bar.



In the middle (see above) is the main area where you lay out and connect blocks to create a 'map'. A 'map' is just a set of connections that describe how inputs from your sensors are converted into sound or light output. You can zoom in and out of this area using the '+' and '-' buttons on the toolbar.

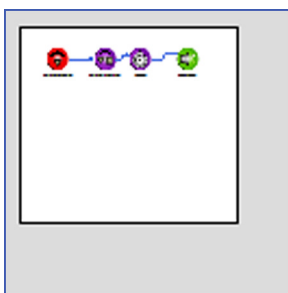
You can open more than one map at the same time, and swap between them by clicking on its tab at the top (just like in an internet browser).

## 4. Info panels

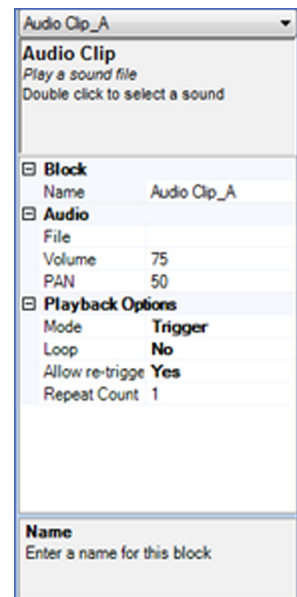
On the right are two panels that give information about blocks, and let you change options quickly. If you have a block selected then various bits of information are displayed:

- The type of block and general description of what it does ('Block info').
- The name (id) of the block, and various options (settings) for it, which you can select and change ('Block settings').
- The help information about the selected block setting ('Block settings help').

At the top of the panel is a 'Block selector'. You can use this to jump quickly to the settings for any block in the map.



At the bottom (see left) is a 'Global view' of your map – a miniature display of the map layout. This is useful if your map is large, or you are zoomed in so cannot see it all in the main map area. A box within the global view shows which part of your map is currently visible, and you can drag this box around in the display.



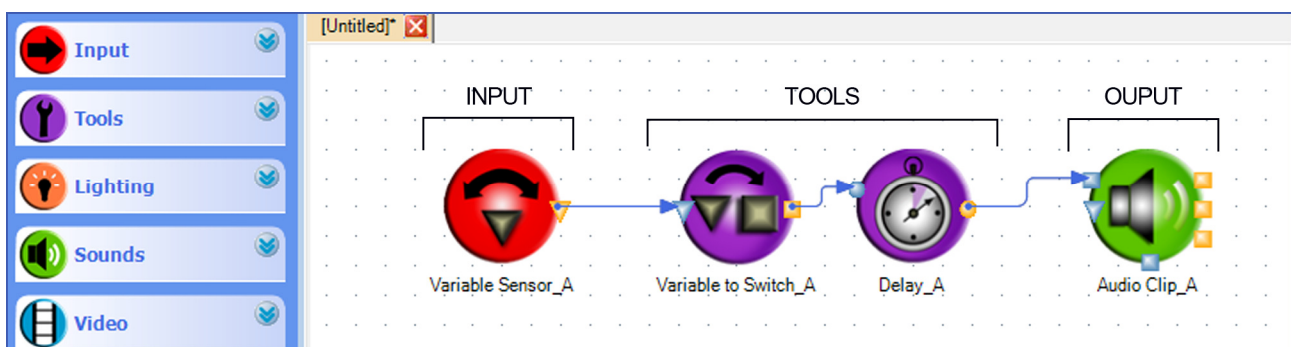
## 2.2.2 Using Designer

The principle of Designer is straightforward: you arrange 'blocks' on-screen, and **link** them together to create a 'map'. The blocks and links specify the way that movement is captured by sensors and converted into sound, light and other outputs.

There are three different types of block:

- 'Input' blocks deal with sensing movement.
- 'Tools' blocks route and change the movement of information.
- 'Output' blocks create sound, light and image in a variety of ways.

The information passed between blocks is called a signal – signals can be a switch action ('on' or 'off'), or variable values (a number between 0 and 100).



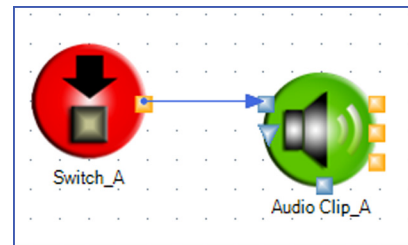
### 2.2.3 Creating a map

To create a map in Designer:

- 1) Find the block you want in one of the block palettes on the left (palettes can be expanded and collapsed by clicking on its title bar). See section 4 ('Blocks Reference') for details of individual blocks – the blocks are colour-coded to match the palettes so, for example, sound blocks will always be coloured green.
- 2) Create the blocks you want by dragging them from the palette onto the Map panel. Designer will automatically tell you if you try to create more than are allowed. All blocks have names, and Designer gives you pre-set ones, which you can change.

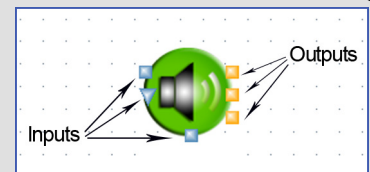
The name and other information for the selected block are shown in the right hand column.

- 3) Each block has a number of settings. You may need to select certain elements for some of these (choosing an audio clip to play, for example), but most settings have default values provided which you can change as required.
- 4) Connect the blocks by dragging a link or wire connection between them. Click on the output of one block, hold the button down and drag the link to the input of another block.

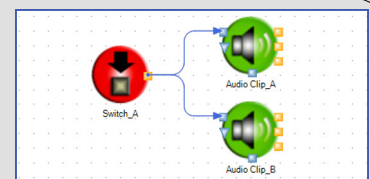


#### RULES FOR CONNECTING BLOCKS

**A block's inputs** are always on the left or bottom, and are blue.  
**A block's outputs** are always on its right hand side, and are yellow.



You can make **one or more** connections from an **output**.  
 You can only make **one** connection to each **input**.



The shape of the connection point indicates the signal you can connect:  
**Triangle** = 'variable' input / output  
**Square** = 'switch' input / output  
**Circle** = either 'variable' **or** 'switch' input / output



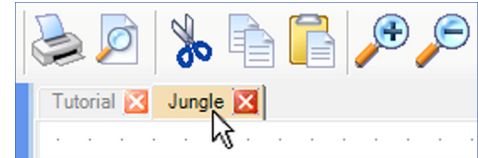
- 5) Press the <Play> button (or F5 key) to open the Player application and perform the current map.



## 2.2.4 Opening, saving and playing maps

You can save your maps to the computer for future use by selecting 'Save' or 'Save As' from the File menu. Files are saved with the suffix ".grf", and can then be opened in Designer and played.

Each map opens in a new tab, so you can quickly change which map you are working with, and copy and paste blocks between maps.



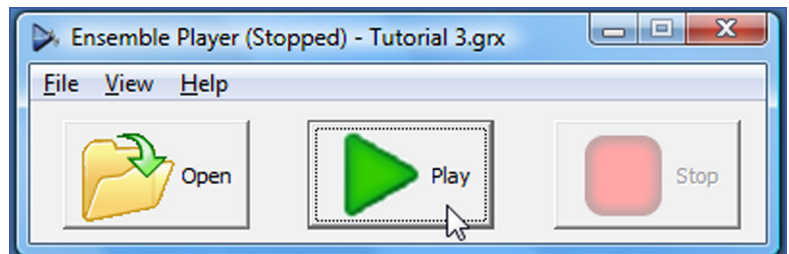
You can work in Designer – make changes to a map, swap to another map (tab), or load in a new map - *while* Player is running. Remember that any changes are *not* reflected in what Player is doing. This is useful, as you can be editing and working on a map, or loading a new one, while performers are still playing (using the previous map).

When you want to audition a new or amended map in Designer, you need to first press the 'Stop' button. Then press the <Play> button again, so that Player restarts and uses the new/changed map.

## 2.3 The Player

When you press the <Play> button (or F5) in the Designer window, the Ensemble Player window opens to run ('play') the map you are currently working with in Designer.

The Player window is actually a separate piece of software. You can use Player on its own as a 'stand alone' system, and open and run your files in it directly. You don't need to use Designer to play files.



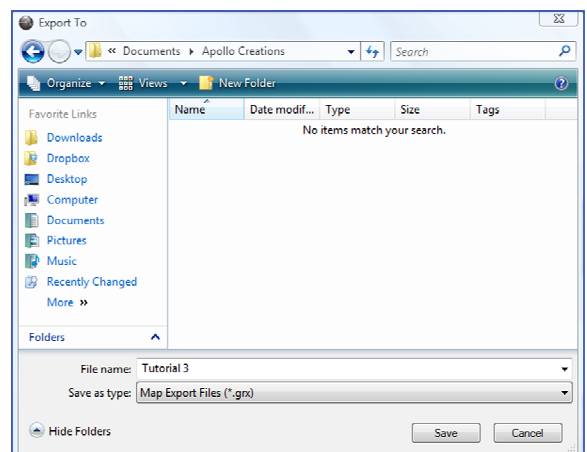
Player makes a far simpler system to give someone else to use, for example to theme a multi-sensory room or exhibition. It also means that changes cannot be made accidentally to carefully constructed maps.

### 2.3.1 Using Player on its own

To operate Player on its own you need to save your map from Designer in a different way - you can't just save a normal map (.grf) file and open it in Player. This is because Player needs more material, for example the audio and image files which may be in use.

This is done by selecting 'Export' from the File menu in Designer (rather than 'Save' or 'Save As').

- Open or create a map (.grf file).
- Select 'Export' from the File menu, and enter a file name to save – choose somewhere different than where your normal Ensemble Designer files are stored. This creates a file with a ".grx" extension.
- Close Designer - you do not need to use it again (although you can leave it open if you want).



You can now operate Player as a stand-alone system:

- Open Ensemble Player (either from the desktop icon or the Windows 'Start' > 'All Programs' menu).
- Press the <Open> button in Player, and select the file you saved earlier. Remember that the file has to be of type .grx which you exported from Designer.
- Press the <Play> button.

Alternatively just double-click the exported map to open it and start playing.

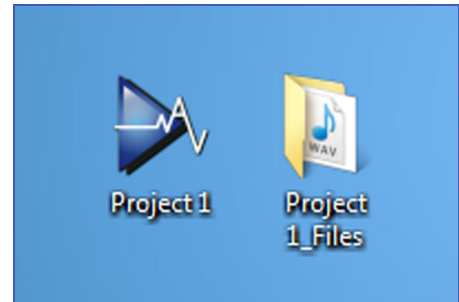
### 2.3.2 Ensemble file types

The method described above also allows you to move Ensemble files between different computers. If you want to move files then you do need to know a little bit more about the files which are saved, so you can copy the correct ones.

Export creates a file with .grx extension (for use with Player) and a folder (with the same name plus '\_Files') containing all the other information.

For example, if we 'Export' our file and name it 'Project 1', then a file 'Project 1.grx' will be saved, but also a folder named 'Project 1\_Files' which will include any audio, video, image and other important files used in the map. If you want to move this to run in Player on another computer, you will need to copy both the file 'Project 1.grx' *and* the folder called 'Project 1\_Files'.

If you wanted someone else to be able to adjust the map you've created, then they would need to 'Import' the 'Project 1.grx' file back into Designer.



.grf and .grx files should appear in Windows with the appropriate icon, allowing you to simply double-click on the file to open the correct application. Sometimes Windows does not register the correct programs to use and you will need to tell it manually what program to use to open the files.

The easiest way to do this is to double-click the file and then select the 'Choose Application' option.

For Designer (.grf), select C:\Program Files\Apollo Creative\Graph Editor.exe

For Player (.grx), select C:\Program Files\Apollo Creative\WinRunner\winrunner.exe

### 3. Creating and running a map in Designer

At this point we will assume that you have the software installed, and an Ensemble16 or EnsembleHub connected to your computer. This section will take you through some basic examples of how to use the software by creating simple maps to make music and sounds. It will also introduce many of the features of blocks and how they work.

In the following tutorials, we will:

- Create a 'Map' in Designer – to specify what we want Ensemble to do.
- 'Play' the map, to run it in Player.
- Press switches or move sensors – sounds will then play in the way we have designed.

The following formatting styles are used for clarity:

- Things you are asked to do begin with >>.
- Block names are bold, e.g. **Audio Clip**.
- Block inputs and outputs are bold and in quotes, e.g. **'Reset'**.
- Categories of block settings are shown underlined, e.g. Playback options.
- Names of block settings are shown in quotes, e.g. 'Loop'.
- Setting values are shown in square brackets, e.g. [No].

Before you start, make sure your sensors and switches are connected into the Ensemble16 or EnsembleHub as follows:

- >> Plug two switches into sockets 1 and 2, ensuring this is in the 'switches' section.
- >> If you have a variable sensor, then plug this into socket 1, but in the 'variable sensors' section.
- >> Start up Ensemble Designer, which will open a blank Map.

Lighting is not included in this tutorial as it is specific to your room or installation.

### 3.1 Tutorial Example 1: Triggering a sound from a switch

First we are going to set-up Ensemble to trigger a single sound by pressing a switch. To do this, we will connect a **Switch** (input) block to an **Audio Clip** (output) block.

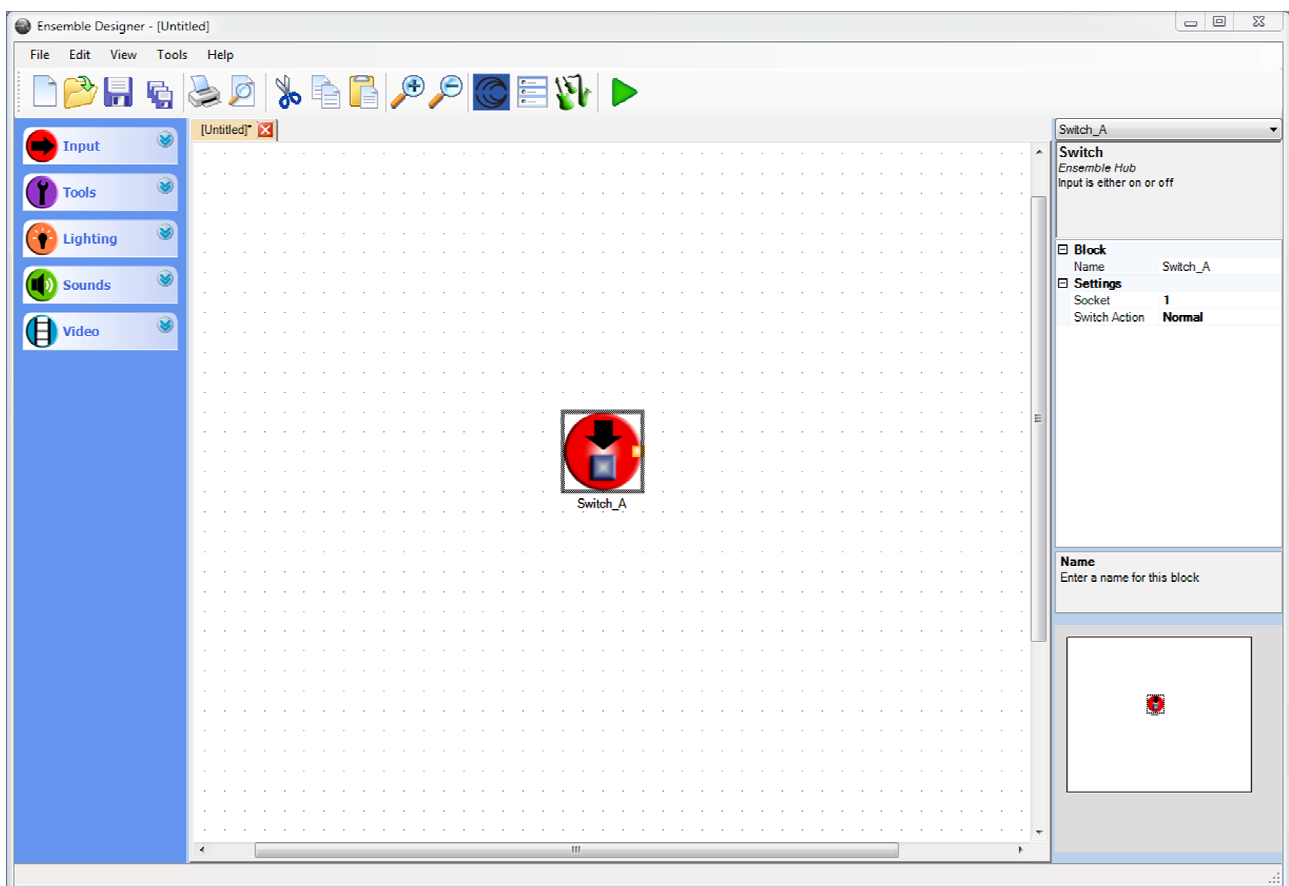
#### 1. Create a Switch block

First we need to create a **Switch** block to correspond to the switch we have plugged into the interface.

>> Expand the 'Input' palette by clicking on its title or the arrows.

>> Drag and drop a **Switch** block onto the map area. You can move it again later by dragging, or selecting and using the cursor keys.

If you want to remove the block from the map, simply click once on the block and then hit the delete key on your keyboard.



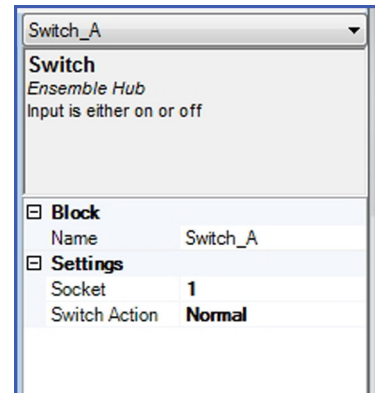
The column on the right hand side shows information and settings for a selected block. Let's look at this **Switch** block.

At the top is general information about this kind of block, in this case telling you it is a 'Switch' block, and is either on or off.

Below this are the settings for this **Switch** block:

(i) Block

'Name': This has been given a default 'Name' of [Switch\_A]. It is advisable to name your blocks clearly to avoid confusion when larger maps are being designed. For example, 'Red Switch'.



You can change this name if you want by clicking in the name box and typing.

(ii) Settings

'Socket': You will see that 'Socket' has been set to 1 by default – this is the number of the actual socket that your switch is plugged into on the interface.

Designer tries to give you sensible default values – for example if you dragged another **Switch** block out of the palette, this would be given a socket number of 2.

You need to make sure the socket number you set here matches up with the actual socket you plug your switch into on the interface. If you move the switch to another socket, then you will need to change the number here. To do this, click on the number, then on the down arrow on the right to select a socket from the menu.

Note that Designer will limit the number of **Switch** blocks you can put on the Map. This is restricted to the number of physical input sockets you have on your interface i.e. eight for the Ensemble16 and four for the EnsembleHub.

'Switch Action': This will be set to [Normal] by default.

[Normal] means the switch will trigger when pressed down (as usual for a switch). If you change this to [Reverse] (click on the setting, then on the down arrow), then the switch would trigger when it is *released*.

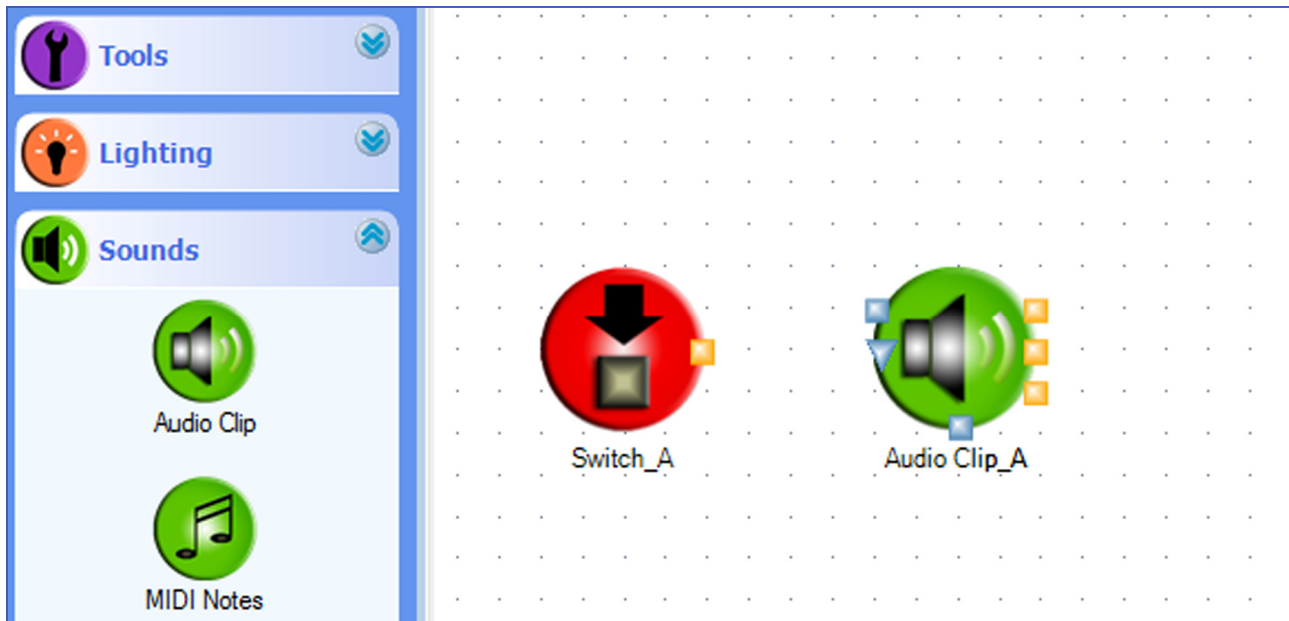


## 2. Create Audio Clip block

Now let's create an **Audio Clip** block, and set it up to play a sound sample.

>> Expand the 'Sounds' palette in the left column by clicking on its title.

>> Drag an **Audio Clip** block onto the map area.



Again, information and settings for this block are shown on the right hand side. Take a few minutes to have a look.

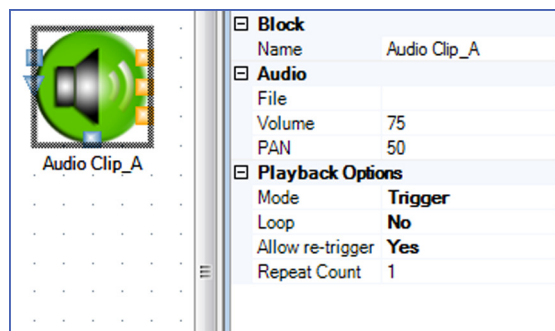
As before, general information about this kind of block is at the top, telling you it is an 'Audio Clip' block and that it plays a wav file. Below this are the settings for the **Audio Clip** block, which are split into three groups:

### (i) Block

'Name': You can see this block has been given a default 'Name' of [Audio Clip\_A].

### (ii) Audio

'File': You will see this is blank at the moment, as you haven't yet chosen a sound file for this block to play.



'Wav' and 'MP3' are types of file format used for audio and sound recordings.

'Volume': Set to 75 by default.

'PAN': Set to 50 by default.

'PAN' sets the position of the sound across stereo speakers:  
0 = fully left, 50 = in the middle, 100 = fully right.

### (iii) Playback Options

'Mode': Set to [Trigger] by default.

[Trigger] means that the sound clip will carry on playing to the end even after the switch is released.

'Loop': Set to [No] by default.

[No] means the clip will not "loop", i.e. will stop after playing through once.

'Allow re-trigger': Set to [Yes] by default.

[Yes] means the sound will be "re-triggered" if the switch is pressed again while the sound clip is playing.  
"Re-triggered" means that the clip will stop immediately and then start playing again from the beginning.

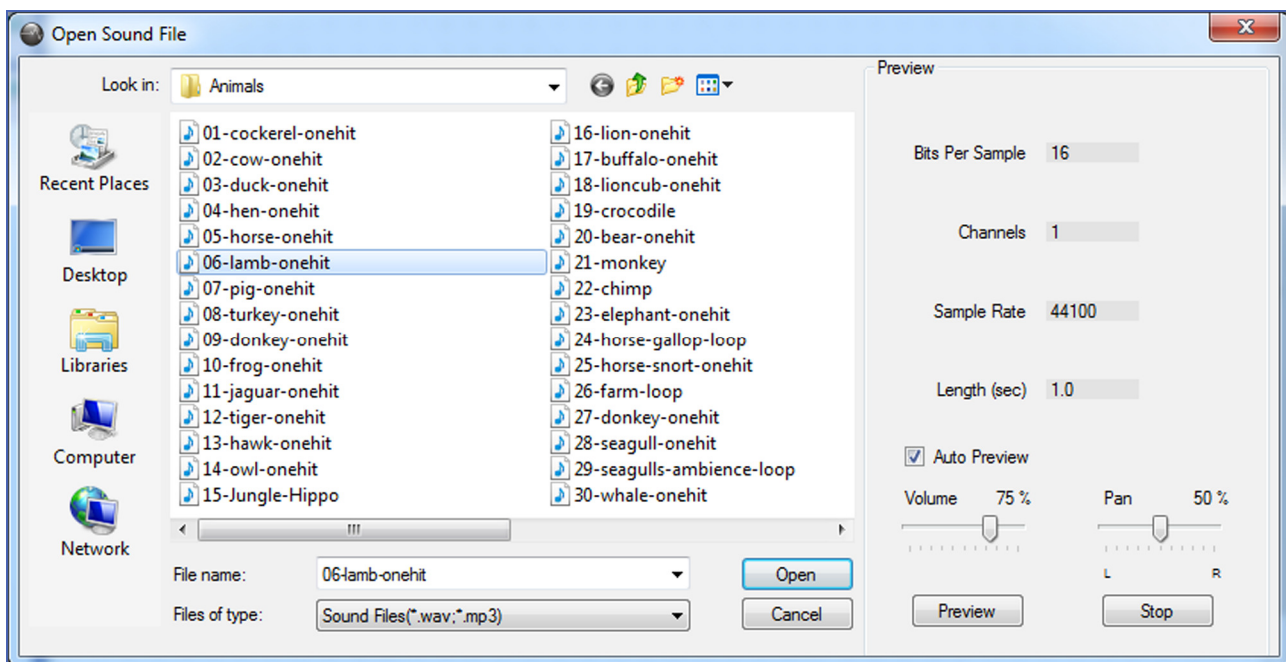
'Repeat Counter': Set to [1] by default.

This is the number of times the clip will repeat by itself before stopping, from once to a maximum of 100 times. This will be greyed out if you have Loop set to [Yes].

Now you need to choose a sound for this **Audio Clip** block to play.

Note that until you select an audio (sound) file, Designer will prompt you to do so whenever you try to change some other setting for the block.

>> Double click on the **Audio Clip** block. This opens a file dialog to choose a sound. You can use any wav or mp3 file from any source, but for this example select a file from the default sounds directory (*C:\Apollo Ensemble\Libraries\Sounds*).



Let's explore this dialog a little before opening a file:

>> Click on wav files to select them (do not double click yet). Information about the file is shown at the top right of the window, such as its length in seconds.

>> Try pressing the 'Preview' button to hear each file.

>> Select 'Auto Preview' and then click on different files to hear them without having to select Preview each time.

>> You can press 'Stop' if the file is a long one.

>> You can try out different volume and pan settings using the sliders. When you have selected a sound these will automatically be transferred into the block's settings (see Step 2(iii)).

>> Finally, find and select the file 'Libraries\Sounds\Farm Animals\06-lamb-onehit.wav'. Load it into the **Audio Clip** block by selecting it and pressing the 'Enter' key, or double click.

Now let's change two of the block's settings, in the Playback Options section:

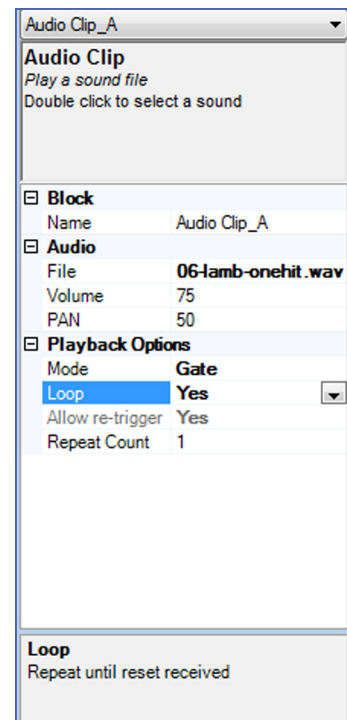
>> Click on 'Mode'.

A downward arrow will appear on the right – click on that to open the menu.

>> Select [Gate]. This means the sound will stop when the switch is released. Imagine that the switch is holding a “gate” open for the sound to come through.

>> Similarly, select 'Loop'.

>> Set to [Yes]. This means the sound clip will continually repeat, until the switch is released.



### 3. Connect Switch block to Audio Clip block

All that remains now is to connect the **Switch** block to the **Audio Clip** block, so that pressing the switch will trigger the sound. We need to connect the “output” of the **Switch** block to an “input” of the **Audio Clip** block.

First, let’s look for the correct input and output to connect. The output of the **Switch** block is easy to see – there is only one. Remember that outputs are always on the right hand side of blocks, and are coloured yellow.

The **Audio Clip** block has 2 inputs. Note that inputs are always on the left hand side and/or bottom of blocks and are coloured blue. The input we want is the top one. Hover the mouse cursor over it, and a pop up will confirm this is the ‘**Trigger**’ input.

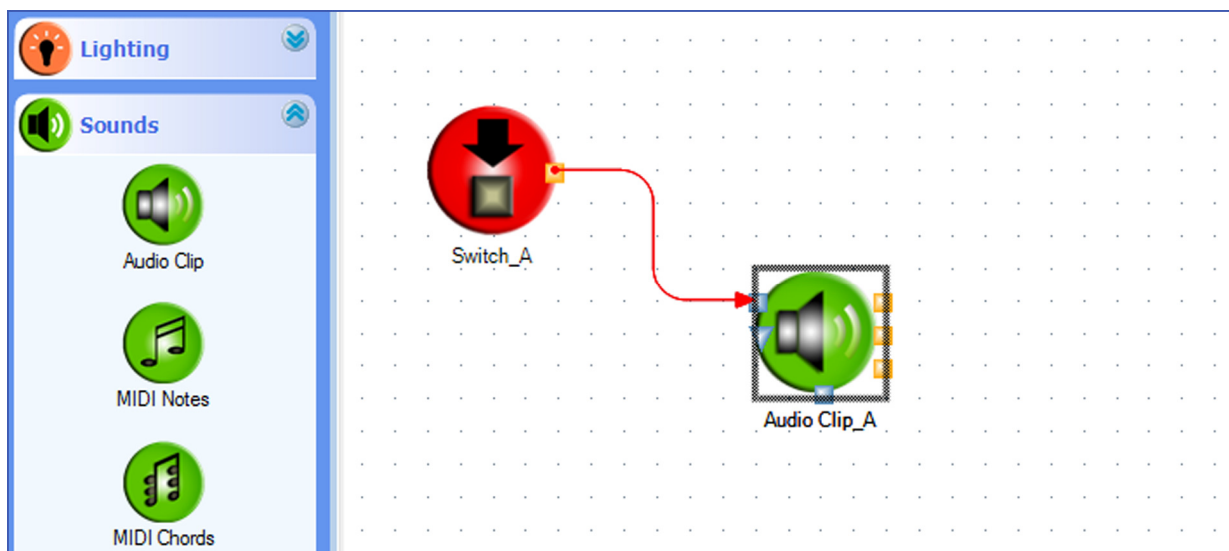
Notice that this input is a square shape - this indicates it is a ‘switch’ type of input (can only be ‘on’ or ‘off’).  
A switch *input* has to have a connection from a switch *output* - such as the one on the **Switch** block  
(notice this is also a square).

Now make the connection:

>> Hover over the output of the **Switch** block until the cursor changes to a ‘hand’, to indicate you can start dragging.

>> Click and drag from this output to the top input of the **Audio Clip** block. When you are over the target input, the cursor will change into a ‘hand’ again and a popup will confirm which input it is (remember we want the ‘**Trigger**’ input).

You should now have created a link from output to input.



You can delete any link by clicking on or near the link to select it, then pressing the ‘Delete’ key.  
Alternatively right-click and select ‘Delete’ or use the Edit menu option.  
You can also move the routing of a link by dragging its vertices (places where the lines change direction).

#### 4. Save

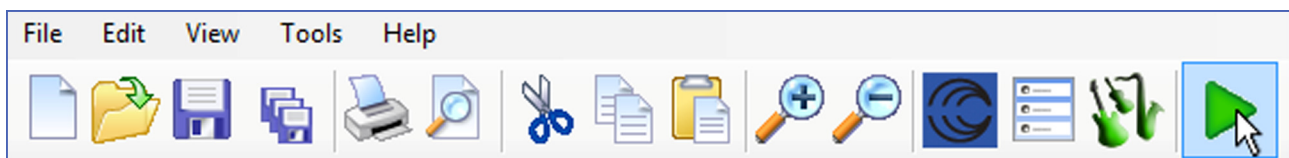
>> Save your map by selecting 'Save' from the file menu (or shortcut Ctrl- S). As it is a new map, you will be asked for a name.

You don't need to save your map to play it, but it's a good habit to get into.

#### 5. Play

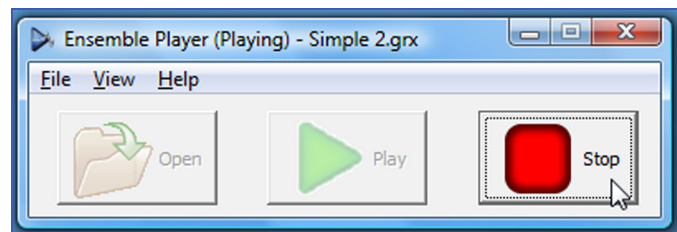
Finally, let's play our simple map and make some sound.

>> Click on the green <Play> button on the right hand end of the tool bar (or press the F5 key).



This will open a new window - the 'Ensemble Player'. See section 2.3 for more details.

You are now in a 'live performing' mode.



Notice that the 'Play' button in Designer changes to a 'Stop' button whilst the Player is active.

>> Press the switch you have plugged into socket 1. The audio clip should start, and continue looping. When you release the switch it should stop.

>> Click on the red <Stop> button on either the Designer or Player to exit, i.e. to stop 'live performing'.

There is no need to close the Player window – in fact, leaving it open will make auditioning changes far quicker.



## 6. Making some changes

Finally, let's try making some changes, to show how quick and easy it can be.

### (a) Turn looping off

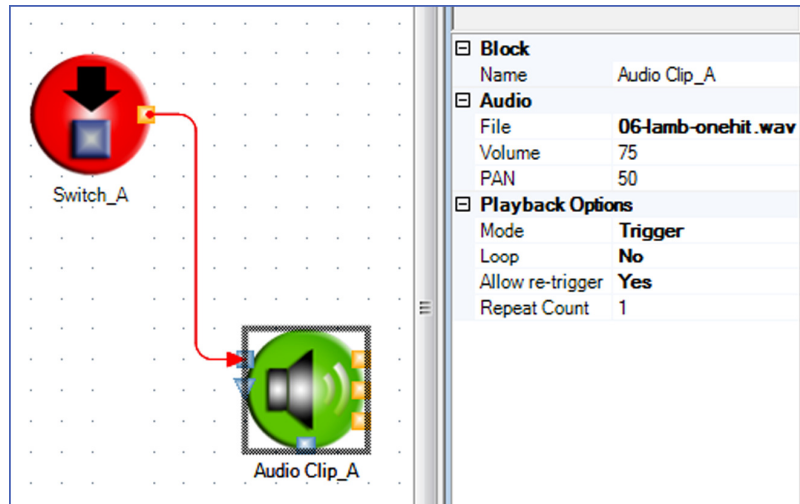
>> Go back to the Designer window.

>> Make sure the **Audio Clip** block is selected and change the 'Loop' setting to [No].

>> Now play the map again (click <Play> button or press F5).

Your Player window may be hidden behind Designer but it will still run.

>> Again, press the switch you have plugged into socket 1. Try doing quick presses or long ones.



Notice that this time the clip only plays once, even if you keep the switch held down. If you press and release the switch quickly the clip will start then stop.

### (b) Change to trigger mode

>> Go back to the Designer window, but leave Player playing.

Notice that you can still edit and create maps while Player is running.

>> Again go to the **Audio Clip** block settings, and change the 'Mode' setting to [Trigger].

Now we need to tell Player to use the new settings – by stopping it, then starting it again.

>> Press the red <Stop> button in Designer. This will change to the 'Play' button again.

>> Press the <Play> button.

You could also have used the 'Stop' and 'Play' buttons in the Player window.

This time pressing the switch will start the clip playing, but releasing the switch will *not* stop it – the clip continues to play through until its end.

For more details on how all these modes work and interact in the **Audio Clip** block, see the Blocks Reference (Section 4).

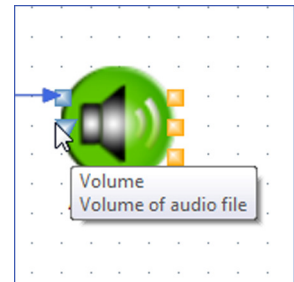
### 3.2 Tutorial Example 2: Changing the volume using a second switch

The **Audio Clip** block has a default 'Volume' and 'Pan' set in the block settings, but you can also change this live whilst playing.

We are now going to use the second switch (the one plugged into socket 2) to change the volume of the sound clip from low to high. To do this we will create a second **Switch** block, and also use a 'Tool' block between the input and output blocks.

>> First, have another look at the **Audio Clip** block - it has a second input below the one you have connected to, which is a triangle shape.

>> Hover the mouse over this input to see the pop up information.



This '**Volume**' input controls the volume of the sound clip whilst playing (and will override its default volume setting). The triangle shape is important: it tells us that this input is expecting a *variable* value (a range of values from 0-100), rather than the simple switch on/switch off action we have used so far.

We could use a sensor to provide this variable value (see Tutorial 3), but in this example we will use a switch to give us *two* different values – let's say 90 and 40 – to choose a high and low volume.

To sum up, when a switch is pressed we want to send the value 90 to the volume, and when not pressed to send the value 40. To do this, we will use a **Switch** block, and then *convert* its on/off output into a *variable* value. This is done using a **Switch to Variable** block from the 'Tools' palette.

#### 1. Create another 'Switch' block

>> Expand the 'Input' palette.

>> Drag another **Switch** block onto the map area (place it under the previous **Switch**).

Have a look at the settings for this block in the right hand column. Notice that by default the 'Name' of your new **Switch** block is set to [Switch\_B], to distinguish it from the first Switch block you have there. Again, you can change this name if you want.

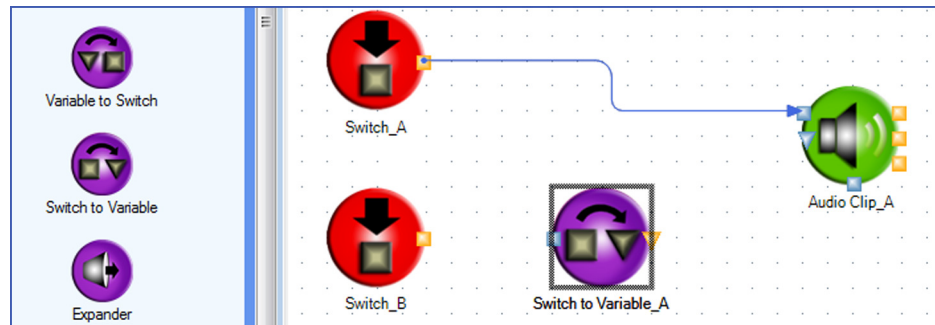
Also notice that by default the **Switch** block's 'Socket' number is set to 2, corresponding to socket 2 that your actual switch is plugged into on the Ensemble interface.

## 2. Create a 'Switch to Variable' block

>> Expand the 'Tools' palette. Scroll down until you see the **Switch to Variable** block.

>> Drag a **Switch to Variable** block onto the map area (place it between the **Switch** and the **Audio Clip**).

Your map should now look something like this:



Making sure that the **Switch to Variable** block is selected (click on it if not), look at the information and settings on the right:

The top help panel tells you what this block does. You set it with two values, for ON and OFF. When the block is triggered by a switch 'on' signal, it will send its 'ON value', otherwise it will send its 'OFF value'.

Switch to Variable	
Converts switch into two variable levels. Set an On and Off %, which are sent when the trigger input is connected to a switch.	
<b>Block</b>	
Name	Switch to Variable_A
<b>Output levels</b>	
ON value	100
OFF value	0

Let's now set up these values:

>> In the Output levels section, click 'ON value'. You can enter numbers by typing in directly.

>> Type 90 and then press Enter.

>> Similarly click 'OFF value' - but this time, try out another way of entering numbers.

>> Click on the downward arrow to open a slider.

>> Drag the slider to set a value of 40%. You can use cursor keys to make fine adjustments.

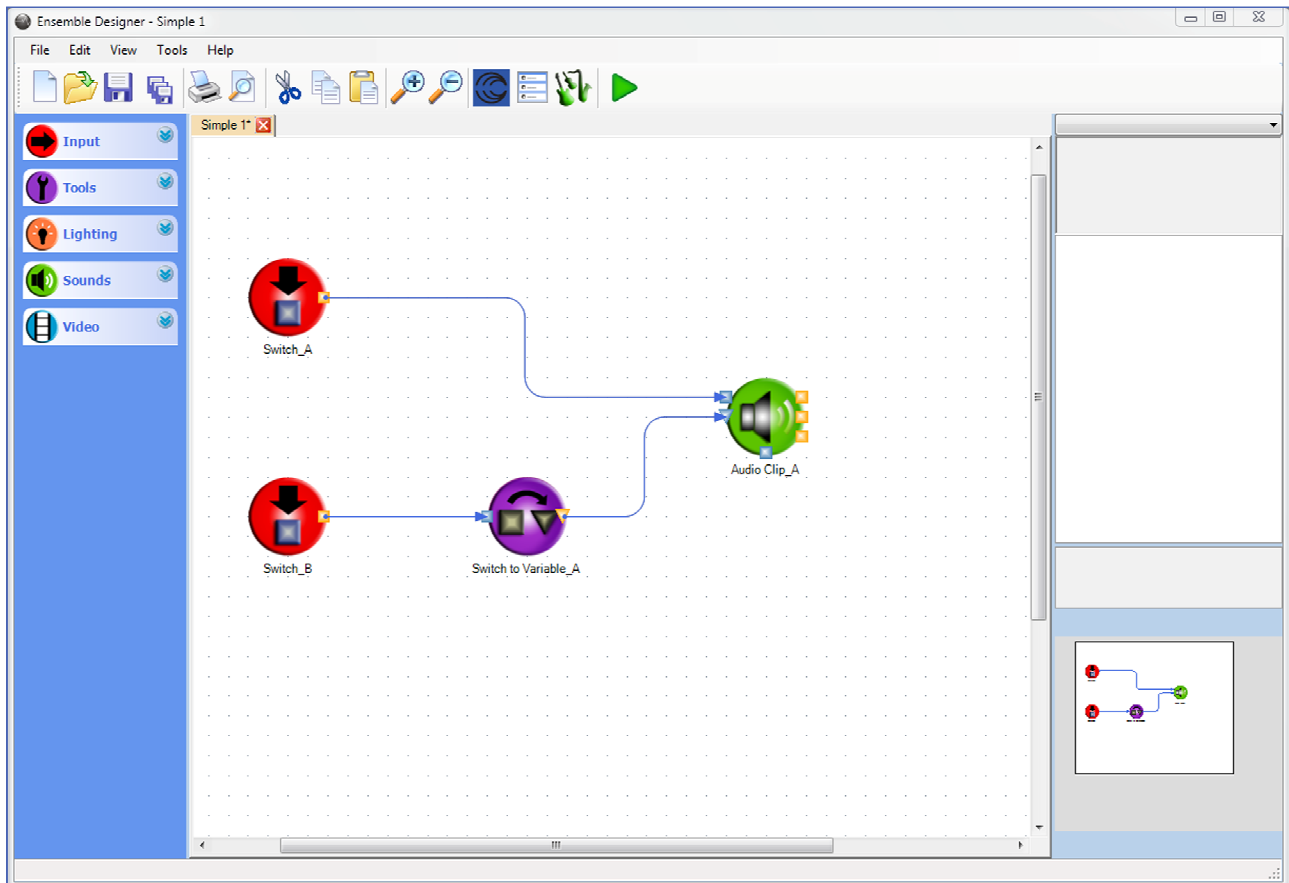


>> Click on the 'Tick' box, or press Enter to finish.

### 3. Connect blocks

>> Connect the output of the **Switch** block to the input of the **Switch to Variable** block.

>> Connect the output of the **Switch to Variable** block to the second input of the **Audio Clip** block, called **Volume**.



### 4. Save and Play

After saving, let's play the map.

>> If you are playing at the moment, first press the <Stop> button.

>> Press the <Play> button (or F5).

>> Press and hold switch 1 (the one plugged into socket 1). The sound clip will play at low volume.

>> When you press and hold switch 2, the volume jumps to the higher level, and the clip will sound much louder when pressing switch 1.

>> Releasing switch 2 reverts back to the lower volume.

### 3.3 Tutorial Example 3: Using a variable sensor

If you have a variable sensor, then we can use this to change the volume of the sound clip continuously.

Let's create another map.

>> From the file menu select 'New'. This will open a new tab with a blank map.



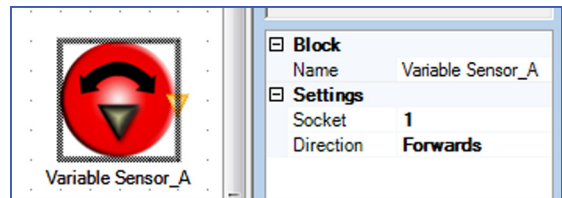
#### 1. Create 'Variable Sensor' block

>> Go to the 'Inputs' palette (see tutorial 1, step 2) and drag a **Variable Sensor** block onto the map area.

Let's look at the settings for this **Variable Sensor** block:

'Socket': Just as we saw for the **Switch** block, this is set to 1 by default, and corresponds to a sensor plugged into socket 1 on the Ensemble16 or EnsembleHub.

Remember this is socket 1 in the "Variable Sensors" section of the interface.



'Direction': This will be set to [Forwards] by default.

[Forwards] means the sensor will give highest values when you are usually closest to the sensor. If you change this to [Backwards], then this is reversed; the highest values will be when you are furthest away from the sensor.

#### 2. Create 'Switch' block

>> Create a **Switch** block, just as you did in tutorial 1 (step 1).

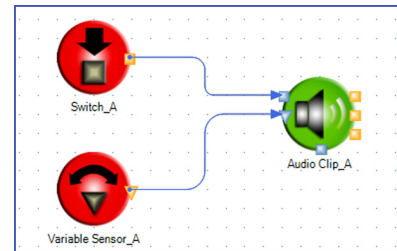
#### 3. Create 'Audio Clip' block

>> Create an **Audio Clip** block and assign a wav file, as before (tutorial 1, step 2).

#### 4. Make connections

>> Connect the **Switch** block (output) to the '**Trigger**' input of the **Audio Clip** block.

>> Connect the output of the **Variable Sensor** block to the '**Volume**' input of the **Audio Clip** block.



#### 5. Save and Play

After saving, play the map – as always if you are already playing, remember to press the <Stop> button first.

>> Press the <Play> button (or F5).

>> As before, pressing and holding switch 1 (socket 1) will start the sound clip playing. Remember you will not hear the sound immediately if the volume is at zero.

>> When you move using the sensor in socket 1, the volume should change up and down.

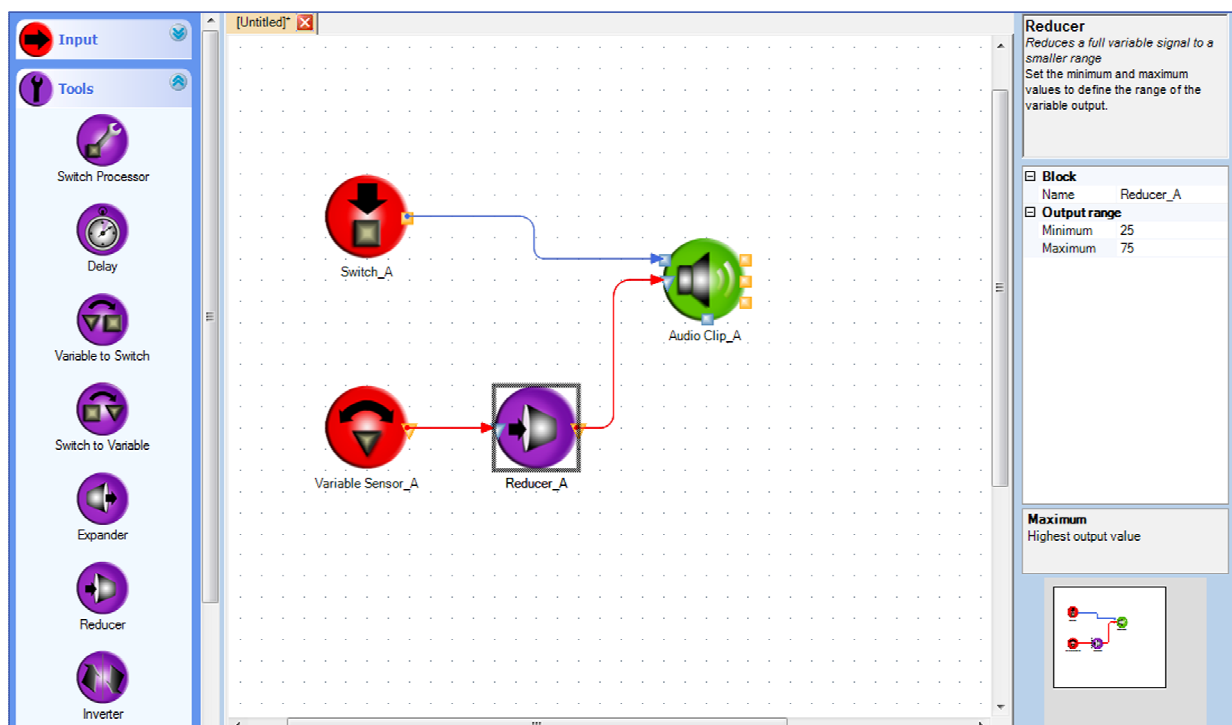
#### 6. Adjust the volume changes

You will probably find the range of volume change is rather too much – it depends on the values the sensor is sending. We can improve this using the **Reducer** block:

>> Place a **Reducer** block on the map (find it in the 'Tools' palette).

>> Connect it between the **Variable Sensor** and the **Audio Clip** blocks (you will need to delete the existing connection).

>> Try setting its 'Minimum' and 'Maximum' in the Output range section to give a narrower and more controllable volume change – the example shown below is for a range of 25 – 75%.



>> Save and Play again.

This time the volume can only go as low as 25% and as high as 75%, even though you can still move in the full sensor range.

### 3.4 Further work

As with any software, the best way to get to know it is to explore and have a play using all the different options.

Explore changing further settings to alter how the blocks behave in all the tutorials. You can also try out other blocks, referring to the Blocks Reference (section 4) and see what happens. Try substituting a different block for each of the blocks you now have, or inserting various 'Tool' blocks between input and output blocks.

If you do something wrong, such as forgetting to set a wav sound for an **Audio Clip**, Designer will tell you and highlight the offending block in red. If you have just asked Designer to play the map, then it will continue to launch Player after warning you of any errors.

Remember to save regularly and enjoy.

---

## 4. Blocks Reference

---

### 4.1 About Blocks

#### 4.1.1 Block palettes

In Ensemble Designer blocks are contained in a number of 'palettes', which will vary depending on the version of the software you have:



**Input** – blocks which represent the actual sensors and switches connected to the Ensemble16 or EnsembleHub interface. Each Input block sends a signal (either on/off or variable values) from one of these into the map.



**Tools** – blocks which change the behaviour, value or direction of signals. These can let us control sound and light in more interesting ways.



**Lighting** – blocks which control DMX lights and other effects.



**Sounds** – blocks which create and control music and sound using MIDI instruments or audio samples.



**Video** – blocks which trigger image and video clips to appear on a screen.

Custom versions of the Apollo Ensemble may include additional palettes, containing specialised blocks for a specific room or environment.

---

#### 4.1.2 Block names

All blocks have a unique *name* which is shown beneath the block's icon on the map window.

By default each block is given a name made up of the *type* of block, e.g. 'Switch', then letters: A, B, C etc. to distinguish them. For example, if you created two **Switch** blocks, their default names would be 'Switch\_A' and 'Switch\_B'. You can edit the name by clicking and typing on it in the block settings panel.

Apart from its name, each type of block has different parameters and settings. These are detailed in the following sections.



### 4.1.3 Handling blocks in the map

You can do various things with blocks once they are on the map area, for example:

- Select a block (click with mouse) or several blocks at once (hold down Ctrl while clicking, or click and drag a box around them).
- Move blocks about (drag with mouse, or use cursor keys). If 'Snap to Grid' is turned on (in the View menu) blocks will jump between grid points. Turn this off to fine tune their positions.
- Delete blocks (menu, right click on the block or Delete key).
- Copy / cut / paste blocks (menu, right click on the selected block or usual keyboard shortcuts). You can cut or copy blocks from one map, then paste them into a map on a different tab.

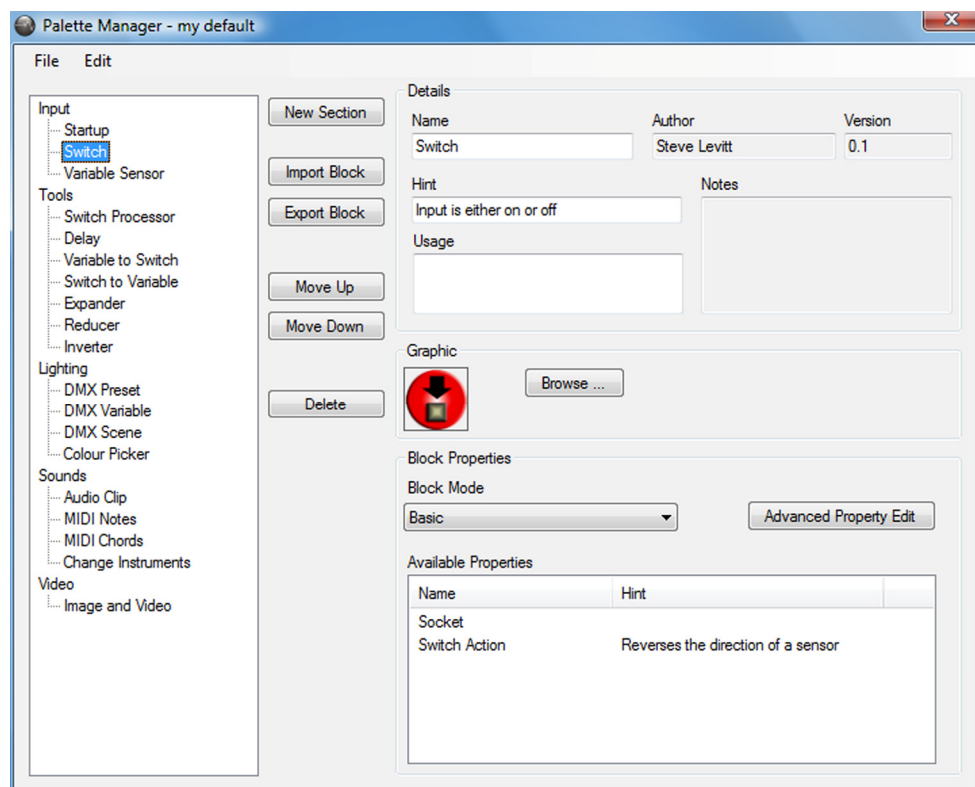
### 4.1.4 Block modes

Blocks can be set to be in one of three different modes using the Palette Manager (Tools > Palette Manager). These modes allow additional inputs, outputs and settings to be enabled for each block.

The three modes are:

- Basic – for everyday use.
- Intermediate (Int) – adding ports and settings that may be required for more complex maps.
- Advanced (Adv) – only for very experienced users.

Settings or ports available in anything other than Basic mode will be indicated in the following section with the abbreviations above.



## 4.2 'Input' Palette



The '**Input**' Palette contains blocks that represent and describe actual sensors and switches connected to the Ensemble interface.

### 4.2.1 'Startup' block



This block sends out a switch 'on' signal as soon as the map starts running (as soon as the 'Play' button is pressed). It acts like a **Switch** block, except that it is 'pressed' automatically when Play starts.

The block has no settings or inputs, and one switch output.

It will only trigger once – and will stay on. Only one **Startup** block is allowed in each map.

### 4.2.2 'Variable Sensor' block



Represents a physical sensor connected to the interface. These sensors detect a range of movements or pressure to produce a varying output value, e.g. by moving nearer or squeezing harder.

A **Variable Sensor** block has a single output, which has a variable value - between 0 and 100%.

The block has two settings:

#### Settings

'Socket'	The socket on the interface where the sensor is connected.
<div>1 – 8 (Ensemble16)</div> <div>1 – 2 (EnsembleHub)</div>	<p>By default, socket 1 is set for the first <b>Variable Sensor</b> block you create. Each further block is then given the next socket number (2, 3, etc.)</p> <ol style="list-style-type: none"> <li>1. Connect your sensor to the same socket on the box as this number.</li> <li>2. Connect your sensor into the set of sockets on the box labeled "Variable Sensors".</li> </ol>
'Direction'	Changes the direction in which the sensor operates - which end is low and which is high; i.e. "which way round" the sensor is.
<div>[Forwards] (default)</div> <div>[Backwards]</div>	<p>The sensor will give highest values when you are closest to it (for example, squeezing harder).</p> <p>The sensor action is reversed, so highest values will be when you are furthest away (for example, not squeezing).</p>



### 4.2.3 'Switch' block



Represents a physical switch connected to the interface.

A **Switch** block has a single output – which is either on or off.

The block has two settings:

#### Settings

'Socket'		The socket on the interface where the switch is connected.
1 – 8 (Ensemble16)		By default, socket 1 is set for the first <b>Switch</b> block you create. Each further block is then given the next socket number (2, 3, etc). 1. Make sure that you connect your switch to the same socket on the interface as this number. 2. Check you connect your switch into the sockets on the box labeled "Switches".
1 – 4 (EnsembleHub)		
'Switch Action'		Changes the direction in which the switch operates.
[Normal]	(default)	The Switch block will trigger 'on' when the switch is pressed down (as is usual for a switch).
[Reverse]		The Switch block will trigger 'on' when the switch is <i>released</i> . When the switch is down it will send an 'off' signal.

#### Important note

As soon as Player starts up, the *current state* of any sensor or switch you have connected is output by the corresponding **Variable Sensor** or **Switch** block. For example, if no switches are being pressed when Player first runs, then each **Switch** block will send an 'off' signal. If that signal is converted to another value (e.g. to an 'OFF Value' by a **Switch to Variable** block), then *that* value is also sent. So, for example, an **Audio Clip** block's volume might be set to a minimum value immediately.



## 4.3 'Tools' Palette

The '**Tools**' Palette contains various blocks that can be used to change the behaviour, or value, of signals:

1. **Inverter** – inverts variable or switch signals so that the output is the opposite of the input.
2. **Switch Processor** – changes a switch on/off signal to make it operate in different ways.
3. **Delay** – delays the transmission of a signal (switch or variable) for a specified time.
4. **Expander** – receives a small variable signal, then 'increases' it to output a different range of values.
5. **Reducer** – receives a variable signal, then 'decreases' it to output a different range of values.
6. **Switch to Variable** – converts a switch into two variable levels.
7. **Variable to Switch** – converts a variable signal to a switch signal when a specified value is reached.
8. **Pulser** – creates a series of 'on' and 'off' pulses whilst it is triggered.

### 4.3.1 'Inverter' block



This block receives a variable or switch signal, then 'reverses' it to output a different range of values.

The block has no settings.

#### Examples

If a **Variable Sensor** was connected to an **Inverter** block, the output would be the opposite number. For example:

IN	OUT	
0	100	For input 0, output is 100.
25	75	Output is 75, because the scale is inverted to run from 100 - 0 whereas the input is from 0 - 100.
80	20	For input 80, output is 20.

For a **Switch**, the output would be the opposite state (on or off). For example:

IN	OUT	
On	Off	If the <b>Switch</b> is pressed, the output would be an off signal.
Off	On	When the <b>Switch</b> isn't being pressed, the output would be an on signal.



### 4.3.2 'Switch Processor' block



This block receives a switch signal and then changes it to make it operate in different ways.

It has a single input, which receives the switch 'on' or 'off' signal (e.g. from a **Switch** block). It then sends out another switch signal ('on' or 'off') in different ways, depending on its settings.

The block has three settings:

#### Settings

'Mode'		Changes the way the block acts. There are 3 modes:
[Off]	(default)	Makes the block have no effect – the output is the same as the input. A connected <b>Switch</b> block will behave just as normal, as if the block is not there. This saves you having to change the connections and bypass the block if you wanted to 'turn it off' temporarily.
[Toggle]		Each time the block is triggered, the output <i>alternates</i> ('toggles') between on and off. Thus, a normal on/off action is converted into a 'push on / push off' action.  <u>Example:</u> A <b>Switch</b> block is connected to the input of the <b>Switch Processor</b> . When the switch is pressed (on), the <b>Switch Processor</b> output goes on. When the switch is released (off) the <b>Switch Processor</b> output <u>stays</u> on. When the switch is pressed a second time, the <b>Switch Processor</b> output goes off. This action repeats.
[Timed]		When triggered, sends out an 'On' signal for a specified time, after which it sends an 'Off' signal. Any switch presses in the intervening time are ignored (depending on the setting of 'Re-Trigger').
'Duration'		When 'Mode' is set to [Timed], then this is the length of time for which the block sends its 'On' value, i.e. how long it stays on for when triggered. NB. This setting is not relevant in other modes, so is shown 'greyed out'.
'Re-trigger'		Controls what happens if 'Mode' is set to [Timed] and the input is triggered again before the 'Duration' has finished (i.e. whilst the timer is running). NB. This setting only matters if the 'Mode' is set to [Timed] and will be shown greyed out in other modes.
[Yes]		The timer can be "re-triggered", so receiving a trigger ('on' signal) will reset the timer to start again from the beginning.
[No]	(default)	The trigger is ignored – the timer will continue until it finishes.



### 4.3.3 'Delay' block



This block delays the transmission of a signal (switch or variable) for a specified time.

It has a single input, which can receive switch or variable signals. It then sends them out again after the preset delay time.

The block has one setting:

#### Settings

'Delay Time'		
1 – 100	(default = 1)	Length of time in seconds that the signal will be delayed for, in 0.1 second intervals.

### 4.3.4 'Expander' block



This block receives a small variable signal and then increases it to output a different range of values. 'Minimum' and 'Maximum' values are set for the block and correspond to an output of 0 and 100. Other values received between the minimum and maximum will be scaled to be within 0 to 100.

The block has two settings:

#### Input range

'Minimum'		Sets the minimum input value, which corresponds to 0.
0 – 100	(default = 25)	
'Maximum'		Sets the maximum input value, which corresponds to 100.
0 – 100	(default = 75)	

For example, if 'Minimum' is set to 40 and 'Maximum' to 90, then:

IN	OUT	
40	0	For input 40, output is 0.
65	50	Output is 50 (half way between 0 and 100), because input 65 is half way between 40 and 90.
90	100	For input 90, output is 100.



In fact, where the block's input is from another block – such as the **Switch to Variable** – a *narrower* range than 0-100 may be received by the block. In this case the output values will be higher / lower than the 'Minimum' and 'Maximum' values set.

For example, if 'Minimum' and 'Maximum' values are again set to 40 and 90, but the input range is only 10-30, then the output will only range from 45-55 (*not* 40-90):

IN	OUT
10	45
30	55

You can also set the 'Maximum' and 'Minimum' values to be 'the wrong way round' - 'Minimum' can be higher than 'Maximum'. This makes the block *invert* the input, so that as the input signal gets higher, the output gets lower.

For example, if 'Minimum' is set to 90 and 'Maximum' to 40, then:

IN	OUT	
40	100	For input 40, output is 100, as set.
65	50	For input 65, output is 50 (half way between 100 and 0).
90	0	For input 90, output is 0, as set.



#### 4.3.5 'Reducer' block



This block receives a variable signal, then 'reduces' it to output a different range of values. A 'Minimum' and 'Maximum' are set for the block. These will be sent out if input values of 0 and 100 are received respectively. Other values received between 0 and 100 will be scaled to be within this range.

The block has two settings:

##### Output range

'Minimum'	Sets the output value to be sent, if an input of 0 is received.
0 – 100 (default = 25)	
'Maximum'	Sets the output value to be sent, if an input of 100 is received.
0 – 100 (default = 75)	

For example, if 'Minimum' is set to 40 and 'Maximum' to 90, then:

IN	OUT	
0	40	For input 0, output is 40.
50	65	Output is 65 (half way between 40 and 90), because input 50 is half way between 0 and 100.
100	90	For input 100, output is 90.

In fact, where the block's input is from another block – such as the **Switch to Variable** – a *narrower* range than 0-100 may be received by the block. In this case the output values will be higher / lower than the 'Minimum' and 'Maximum' values set.

For example, if 'Minimum' and 'Maximum' values are again set to 40 and 90, but the input range is only 10-30, then the output will only range from 45-55 (*not* 40-90):

IN	OUT
10	45
30	55

You can also set the 'Maximum' and 'Minimum' values to be 'the wrong way round' - 'Minimum' can be higher than 'Maximum'. This makes the block *invert* the input, so that as the input signal gets higher, the output gets lower.

For example, if 'Minimum' is set to 90 and 'Maximum' to 40, then:

IN	OUT	
0	90	For input 0, output is 90.
50	65	For input 50, output is 65 (half way between 90 and 40).
100	40	For input 0, output is 40.





#### 4.3.6 'Switch to Variable' block



This block converts a 'switch' signal into two 'variable' signal levels.

It receives a switch 'on' or 'off' signal, and outputs two different 'variable' values – these can be sent to another block which is expecting variable values. An 'ON value' and 'OFF value' are set, which will be sent out respectively if a switch 'on' or 'off' input is received.

The block has two settings:

##### Output levels

'ON Value'		The value to be sent out, if an 'on' switch signal is received.
0 – 100 (default = 100)		
'OFF Value'		The value to be sent out, if an 'off' switch signal is received.
0 – 100 (default = 0)		



#### 4.3.7 'Variable to Switch' block



This block receives a variable signal (e.g. from a **Variable Sensor** block) and converts it into a switch 'on' or 'off' signal.

The **Variable to Switch** block has two values set for 'Lower Trigger' and 'Upper Trigger'. It receives a variable signal and when this crosses its 'Lower Trigger' value, the block outputs a switch 'on' signal. Similarly, when the input value crosses its 'Upper Trigger' value, the block outputs a switch 'off' signal.

The input has to *cross* the value, *not* just reach it – thus having the Upper Trigger value set to the default 100 means that the switch 'off' signal will *not* be sent when the input reaches 100.

The block has three settings:

##### Trigger points

'Lower Trigger'		The input value at which the switch will be 'turned on'. With the 'Direction' setting (see below) set to its default [Forwards], then an 'on' switch signal will be sent out when the variable sensor input passes above this value.
0 – 100 (default = 50)		
'Upper Trigger'		The input value at which the switch will be turned off. With 'Direction' again set to its default [Forwards], then an 'off' switch signal will be sent out when the variable input passes above this value.
0 – 100 (default = 100)		
'Direction'	(Int/Adv)	Changes the direction that the switch output operates.
[Forwards]	(default)	The switch goes 'on' when the input goes above the lower value, and 'off' when it goes above the upper value. It is thus 'on' then the input is between lower and upper values.
[Backwards]		The switch output will go 'on' when (a) the input goes <i>below</i> the lower value, or (b) when it goes above the upper value. It is thus 'off' when the input value is <i>between</i> the lower and upper values, and 'on' when it is outside this range. This can give a useful "turn on at either end" operation for a variable sensor NB. If the variable sensor is in the zero position when Player starts, then a switch 'on' signal will immediately be sent.



#### 4.3.8 'Pulser' block



This block receives a switch signal and then sends out a series of 'on' and 'off' pulses whilst it is triggered.

It has a single input, which receives the switch 'on' or 'off' signal (e.g. from a **Switch** block). It then sends out a series of switch signals ('on' or 'off').

The block has two settings:

#### Settings

'Time On'		Sets the length of the 'on' pulse.
0 – 120 s (default = 4)		
'Time Off'		Sets the length of the 'off' pulse.
0 – 120s (default = 1)		



## 4.4 'Lighting' Palette

The '**Lighting**' Palette contains blocks required to control lighting equipment and special effects using DMX.

1. **DMX Preset** – sends a fixed value on the selected DMX channel when triggered.
2. **DMX Variable** – receives a variable signal and sends this value on the selected DMX channel.
3. **DMX Scene** – sends a set of values on multiple DMX channels when triggered.

These elements will only work if you have the additional USB-to-DMX cable. The element controlled by the DMX channel (brightness, position, colour) will depend on the lights you have selected and their specific settings. Please see Appendix 4 for further details.

### 4.4.1 'DMX Preset' block



This block converts a switch into two 'variable' signal levels to be sent via DMX.

It receives a switch 'on' or 'off' signal and outputs two different 'variable' values on the DMX channel selected. An 'On level' and 'Off level' are set, which will be sent out respectively if a switch 'on' or 'off' input is received.

The block has four settings:

#### Settings

'DMX Channel'	The DMX channel to send the data.	
1 – 512	By default, 'DMX Channel' is set to 1 for the first <b>DMX Preset</b> block you create. Each subsequent block is given the next channel number (2, 3, etc.). The channel here should be matched to the channel on the light or effect you are controlling.	
'On level'	The DMX value to be sent out, if an 'on' switch signal is received.	
0 – 255 (default = 255)		
'Off level'	The DMX value to be sent out, if an 'off' switch signal is received.	
0 – 255 (default = 0)		
'Fade time'	The time it takes to move between the 'On level' and 'Off level' when the block is triggered.	
0 – 10s (default = 0.5)		



#### 4.4.2 'DMX Variable' block



This block receives a variable signal (e.g. from a **Variable Sensor** block) and sends it on the DMX channel selected. The value received (0-100%) is scaled to create a DMX value of 0 to 255.

#### Settings

'DMX Channel'		The DMX channel to send the data.
1 – 512	(default = 1)	By default, 'Channel' is set to 1 for the first <b>DMX Variable</b> block you create. Each further block is then given the next channel number (2, 3, etc).

#### 4.4.3 'DMX Scene' block



This block converts a switch into two sets of DMX values, sent on multiple DMX channels.

It receives a switch 'on' or 'off' signal and outputs two different sets of DMX values sent on the DMX channels selected. An 'On' and 'Off' value are set, which can be sent out respectively if a switch 'on' or 'off' input is received.

#### Settings

'Scene'		The set of DMX data that will be sent.
These are set using a separate editor window		To open the DMX editing window, double click on the block. Alternatively, you can click on the 'Scene' setting, then on the down arrow which appears on the right hand side.
'Mode'		Controls how the block sends DMX data.
[Gate]	(default)	Sends the 'On' values when it receives an 'on' switch signal, and sends the 'Off' values when it receives an 'off' switch signal.
[Trigger]		Sends the 'On' values when it receives an 'on' switch signal. It ignores an 'off' switch signal and will only send the 'Off' values when the block's <b>'Reset'</b> input is triggered.
'Fade time'		The time it takes to move between the 'On level' and 'Off level' when the block is triggered.
0 – 10s		(default = 0.5)



## 4.5 'Sounds' Palette

The '**Sounds**' Palette contains all the blocks required to make music and sound. This includes triggering audio samples and playing using MIDI instruments:

1. **Audio Clip** – play a sound sample (wav or mp3 file).
2. **MIDI Notes** – play one or more notes (using a MIDI instrument), in order or individually.
3. **MIDI Chords** – play one or more chords (using a MIDI instrument), in order or individually.
4. **Change Instruments** – change to a new set of MIDI instruments.
5. **MIDI Controller** – send MIDI Controller information on the specified channel.

### 4.5.1 'Audio Clip' block



Plays a sound file.

Inputs can start and stop playback, and control volume.

The block has 3 inputs:

1. <b>'Trigger'</b> (Switch)	Starts sound playing.
2. <b>'Volume'</b> (Variable)	Alters the sound volume during play.
3. <b>'Reset'</b> (Switch)	Stops playing.

The block has a number of settings, divided into categories:

#### Audio

<b>'File'</b>		The sound file which this block will play.
[File name]		The name of the sound file (wav or mp3 format) assigned to the block.
<b>'Volume'</b>		Sets the volume of the sound file.
0 – 100 (default = 75)		The block's <b>'Volume'</b> input can override this value during play.
<b>'PAN'</b>		Sets the balance between the left and right speakers.
0 – 100 (default = 50)		0 = fully left; 50 = centre; 100 = fully right



## Playback Options

'Mode'		Controls how the sound plays.
[Trigger]	(default)	Starts playing when it receives an 'on' switch signal, and then carries on playing to the end. It ignores an 'off' switch signal. NB. Sound <i>will</i> stop if the block's ' <b>Reset</b> ' input is triggered
[Gate]		Starts playing when it receives an 'on' switch signal, but stops as soon as it receives an 'off' switch signal. Only plays while the switch is 'on'. NB. In this case, the setting below for 'Allow re-trigger' has no effect, so will be 'greyed out'.
'Loop'		Controls how, or if, the sound repeats.
[Yes]		The sound will 'loop', i.e. will repeat playing indefinitely. How the clip is <i>stopped</i> depends on other settings and connections: (a) If Mode is [Gated], then the sound will loop while it receives an 'on' signal, and then stop when it receives an 'off' signal. (b) If Mode is [Triggered], then the sound will continue looping indefinitely - in this case it will only stop when the block's ' <b>Reset</b> ' input is triggered. NB. The 'Repeat Counter' setting (below) is then irrelevant and is shown greyed out.
[No]	(default)	The clip will not 'loop', however the clip <i>can</i> still repeat – if 'Repeat Counter' (below) is set to more than 1.
'Allow re-trigger'		Controls what happens if an 'on' switch signal is received whilst the sound clip is playing.
[Yes]	(default)	The sound can be "re-triggered" – receiving a trigger ('on' switch signal) will interrupt play, and then start the sound again from the beginning.
[No]		The trigger is ignored – the sound will keep playing to the end. NB. This setting only matters if the 'Mode' is set to [Trigger]. If 'Mode' is set to [Gated], then the sound can't be 're-triggered' during playback and this setting will be shown greyed out.
'Repeat Counter'		The number of times the sound will repeat playing before stopping.
1 – 100	(default = 1)	If 'Mode' is set to [Gated], then an 'off' switch signal will <i>interrupt</i> and stop play before this number is reached. If the 'Loop' setting is [Yes] then the clip will loop indefinitely, and this 'Repeat Counter' setting is ignored and is shown greyed out.





#### 4.5.2 'MIDI Notes' block



Plays one or more notes (using a MIDI instrument), either in series or individually.

It contains a set of notes in a sequence (or just a single note). Each note may have a different duration, different instrument and loudness (volume or MIDI 'velocity').

Notes can be triggered in two ways:

1. Play notes one at a time in sequence - if you connect a switch to the **Trigger** input, then each press of the switch will play the next note in the sequence. At the end, it will cycle back to the beginning.
2. Play any note ('random access') – if you connect a variable sensor to the **Trigger** input, then each position of the sensor will trigger the note at the corresponding position, as spaced within the set. Exactly which note depends on how many there are, and the 'trigger point' set for each note - see below for details.

The block has 3 inputs:

1. <b>'Trigger'</b> (Switch or Variable)	Plays appropriate note (depending on input – see above).
2. <b>'Volume'</b> (Variable)	Alter the note loudness (MIDI 'velocity' or volume).
3. <b>'Reset'</b> (Switch)	Resets the sequence of notes back to the beginning.

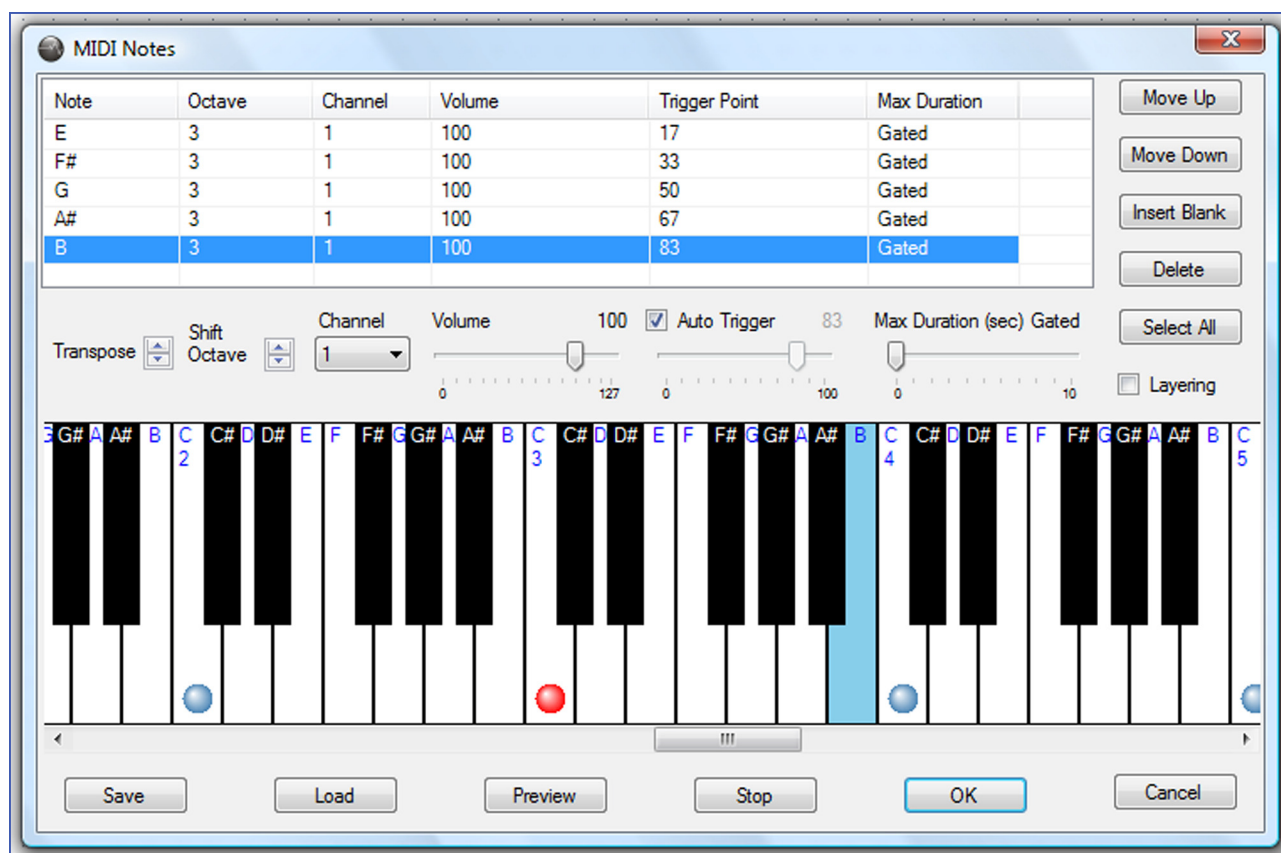
In its settings display, the block only shows some of the names of the notes it contains. All the settings are made in a separate editing window.

#### Notes

'Notes'	The set of notes this block will play.
These are set using a separate editor window	To open the note editing window, double click on the block. Alternatively, you can click on the 'Notes' setting, then on the down arrow which appears on the right hand side.



## Operating the 'MIDI notes' block editor window:



Task (I want to...)	Operation (how to do it...)
Add note to selection	Click on a note on the music keyboard.
Audition note without adding to selection	Ctrl- click on a note on the music keyboard.
Select single note	In top list, click on note. Can also then use Cursor up / down keys.
Select multiple range of notes	In top list, Shift- click on notes. Or use Shift- Cursor up / down keys.
Select multiple individual notes	In top list, Ctrl- click on notes.
Select all notes	Press the 'Select All' button.
Change order of notes within the sequence	Press the 'Move Up' or 'Move Down' buttons to move the selected notes up or down in the list.
Delete selected notes	Press the 'Delete' button, or use the Delete key.
Insert a gap into the note list	Press 'Insert Blank' button. This inserts a 'gap' into the list of notes. NB. This has the same effect as manually changing the trigger points of all the notes above this point in the range.
Change pitch of selected notes	Press the 'Transpose' up or down arrows to change the pitch by one semitone each time.
Change octave of selected notes	Press the 'Shift Octave' up or down arrows to change the octave by one each time.

Set MIDI channel of selected notes	Click on the 'Channel' display to open a menu of channels, then select a channel number. You can also click a second time on the number then type it.
Set volume of selected notes	Drag the 'Volume' slider to set a velocity of 0-127.
Set duration of selected notes	<p>Drag the 'Max Duration' slider. A value between 0.1 seconds to 10 seconds can be set.</p> <p>When triggered, the note will play until released or the duration is reached, whichever is shorter.</p> <p>NB. If the slider is dragged lower than 0.1 secs, then duration will be set to 'gated' – see below</p>
Set duration of selected notes to be controlled by switch on / off	<p>Drag the 'Max Duration' slider fully left. The 'Max Duration' display should then show as 'Gated'.</p> <p>The note will then play indefinitely while the block's '<b>Trigger</b>' switch input is 'on', and will stop when a switch 'off' signal is received.</p>
Set spacing of notes to be automatic	<p>Tick the 'Auto Trigger' box. This then sets each note's trigger point automatically, which will space each note out <i>evenly</i> through the input (variable signal) range. The trigger point for a note is the position within the input signal range 0 – 100 where it will be triggered to play.</p> <p>Any new note added will also automatically have an evenly spaced trigger point set.</p>
Set the spacing of selected notes manually	<p>Untick the 'Auto Trigger' box. Then drag slider (below) to set a trigger point for the notes.</p> <p>NB. If you later tick 'Auto Trigger', any previously manually set trigger points are not remembered.</p>
Set notes to build up in a 'layer'	Tick the 'Layering' box. As the sensor or switch moves through the sequence, notes are <i>added</i> , the previous notes continue playing to create a chordal effect. A 'Blank' note has the effect of switching all the notes off.
Save sequence	Press the 'Save' button, and enter file name
Load sequence	Press the 'Load' button, and select file.
Audition the notes	Press the 'Preview' button.
Exit and save settings	Press the 'OK' button.



### 4.5.3 'MIDI Chords' block



Plays one or more chords, using a MIDI instrument - either in series or individually.

It contains a set of chords in a sequence (or just a single chord). Each chord may have a different duration, and each note within the chord may have a different instrument and loudness ('velocity').

Chords can be triggered in two ways:

1. Play chords one at a time in sequence - if you connect a *switch signal* to the block's '**Trigger**' input, then each press of the switch will play the next chord in the sequence. At the end, it will cycle back to the beginning.
2. Play any chord ('random access') - if you connect a *variable signal* to the block's '**Trigger**' input, then each position of the sensor will trigger the chord at the corresponding position in the set. Exactly which chord is triggered depends on how many there are, and the 'Trigger Point' which is set for each chord - see below.

**Trigger Points:** A chord's trigger point is the position within the range of the input signal (0-100) at which the chord will be triggered to play. For example a trigger point of 50 would trigger the chord when the input value is half way (between top and bottom of its range).

By default, the block will set these positions automatically (using its 'Auto Trigger' feature). However you can turn this off if you want to, and change each chord's trigger point manually, to make some chords be 'nearer' each other in the sensor's movement range.

Chord	Trigger Point	Max Duration
Chord 1	25	Gated
B Major Dominant	50	Gated
Chord 3	75	Gated

Chord Name	<input checked="" type="checkbox"/> Auto Trigger	50	Max Duration (sec)	Gated
B Major Dominant		<input type="text" value="50"/>	<input type="text" value="5"/>	

The block has 3 inputs:

1. ' <b>Trigger</b> ' (Switch or Variable)	Plays appropriate chord (depending on input, see above).
2. ' <b>Volume</b> ' (Variable)	Alter the loudness (MIDI 'velocity' or volume) of the chord.
3. ' <b>Reset</b> ' (Switch)	Resets the sequence of chords back to the beginning.

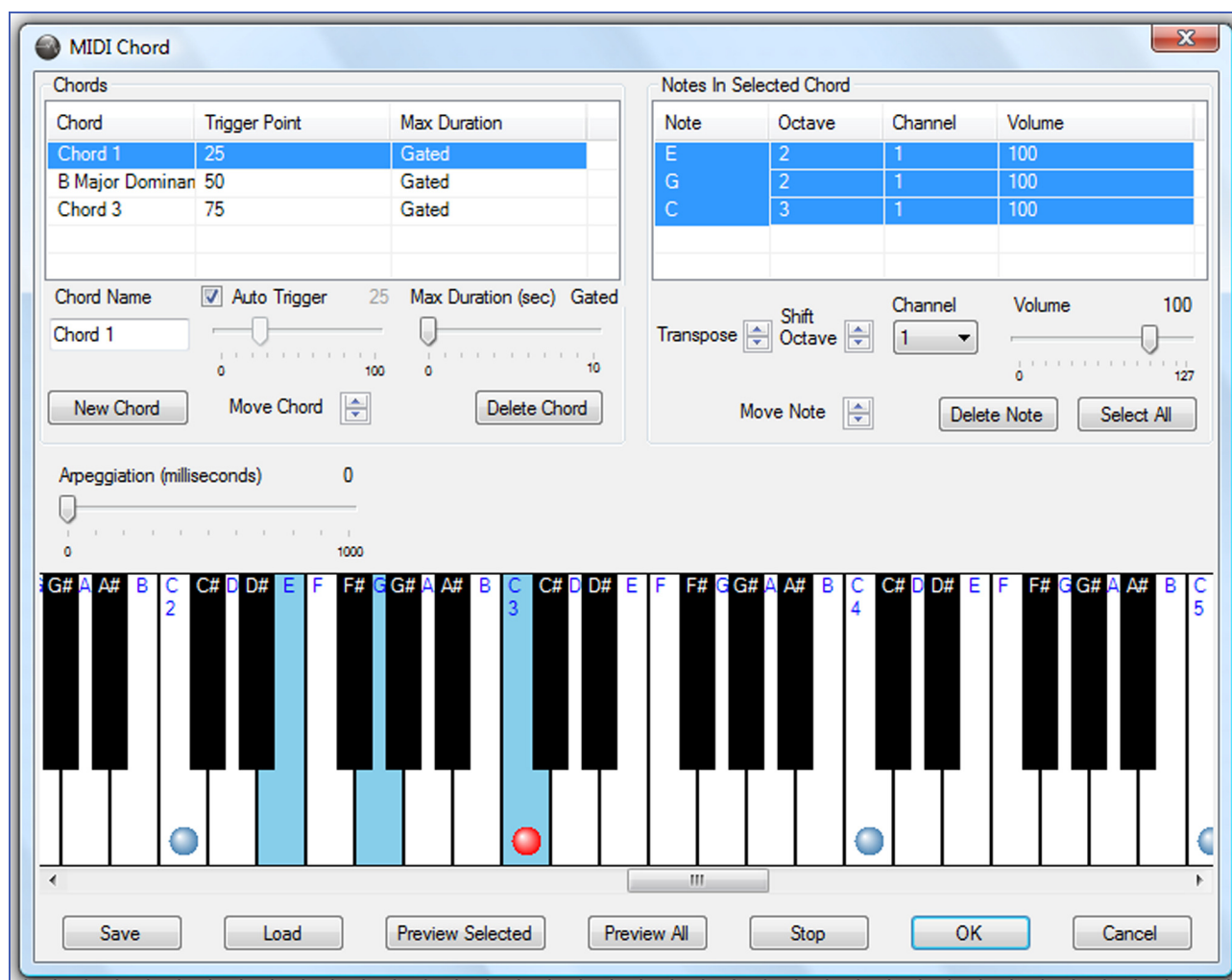
### Chords

In its settings display, the block only shows some of the names of the chords it contains. All the settings are made in a separate editing window.

'Chords'	The set of chords this block will play.
Set using a separate editor window	To open the chord editing window, double click on the block. Alternatively, you can click on the 'Chords' setting, then on the down arrow which appears on the right hand side.



## Operating the 'MIDI Chords' block editor window:



Task (I want to...)	Operation (how to do it...)
<b>Tasks for Chords</b>	
Select a single chord	In top left list, click on a chord. Can also then use Cursor up / down keys.
Select a multiple range of chords	In top left list, Shift- click on a chord. Or use Shift- Cursor up / down keys.
Select multiple individual chords	In top left list, Ctrl- click on chords.
Set duration of selected chords	<p>Drag the 'Max Duration' slider. A value between 0.1 seconds to 10 seconds can be set.</p> <p>When triggered, the chord will play until switch input goes to 'off' or the duration set is reached.</p> <p>NB. If the slider is dragged lower than 0.1 secs., then duration will be set to 'gated' – see below (this is the default setting).</p>

Set duration of selected chords to be controlled by switch on / off	Drag the 'Max Duration' slider fully left. The 'Max Duration' display should then show as 'Gated'.
Change name of chord	The chord will then play indefinitely while the block's <b>'Trigger'</b> switch input is 'on', and will stop when a switch 'off' signal is received.
Set spacing of chords to be automatic	Click in 'Chord Name' box, and type.
	Tick the 'Auto Trigger' box (default). This then sets each chord's 'trigger point' automatically – which will space each chord out <i>evenly</i> through the input (variable signal) range.
	The trigger point is the position within the input signal range 0-100 where a chord will be triggered to play – see above.
	Any new chord added will also automatically have an evenly spaced trigger point set.
Set the spacing of selected chords manually	Untick the 'Auto Trigger' box. Then drag slider (below) to set a trigger point for the chords.
	NB. If you later tick 'Auto Trigger', then any previous 'manually set' Trigger Points are not remembered.
Add a chord	Press the 'New Chord' button.
Change order of selected chord within the list	Press the 'Move Chord' up or down arrows to move the selected chord up or down in the list.
Delete selected chords	Press the 'Delete Chord' button, or use the Delete key.
Make chord arpeggiate	Drag the 'Arpeggiation' slider to set a millisecond value for the time delay between notes in the chord.
	Notes can be spaced up to one second (1000 milliseconds) apart.
	Setting to 0 (default) makes the chord play normally.
Save chords	Press the 'Save' button.
	NB. This will save <i>all</i> the chords in the list, not just those that are selected.
Load chords	Press the 'Load' button, and select a file.
Audition selected chord	Press the 'Preview selected' button.
Audition all chords in turn	Press the 'Preview All' button.
<u>Tasks for Notes <i>within</i> a chord</u>	*** You need to select a <i>single</i> chord before you can do these ***
Add note to a chord	Click on a note on the music keyboard.
Audition note without adding to chord	Ctrl- click on a note on the music keyboard.
Select single note in a chord	In top right list, click on a note. Can also then use Cursor up / down keys.
Select multiple range of notes in a chord	In top right list, Shift- click on notes. Or use Shift- Cursor up / down keys.
Select multiple individual notes in a chord	In top right list, Ctrl- click on notes.
Select all notes in a chord	Press the 'Select All' button.
Change order of notes within a chord	Press the 'Move Note' up or down arrows to move the selected notes up or down in the list. You can also move multiple selected notes.

Delete selected notes from a chord	Press the 'Delete Note' button, or use the Delete key.
Change pitch of selected notes	Press the 'Transpose' up or down arrows to change the pitch by one semitone each time.
Change octave of selected notes	Press the 'Shift Octave' up or down arrows to change the octave by one each time.
Set MIDI channel of selected notes	Click on the 'Channel' display to open a menu of channels, then select a channel number. Alternatively, you can click a second time on the number then type it.
Set volume of selected notes	Drag the 'Volume' slider to set a velocity of 1-127.
Exit and save settings	Press the 'OK' button





#### 4.5.4 'Change Instruments' block



Change the instruments used by the MIDI Chords or MIDI Notes blocks - on any (or all) of the 16 MIDI channels (see Appendix 3).

Various changes can be made:

- Use different MIDI instruments.
- Change the settings for Volume and Pan.
- Change the 'effects' settings for 'Reverb' (simulates being in different kinds of acoustic space, e.g. a large echoing room), and for 'Chorus' (simulates having several instruments playing at once, like an orchestral string section).

This block does not let you reverse these changes. To revert back to the original settings you will need to create a second **Change Instruments** block with the original settings, and then trigger this.

The block has a single input:

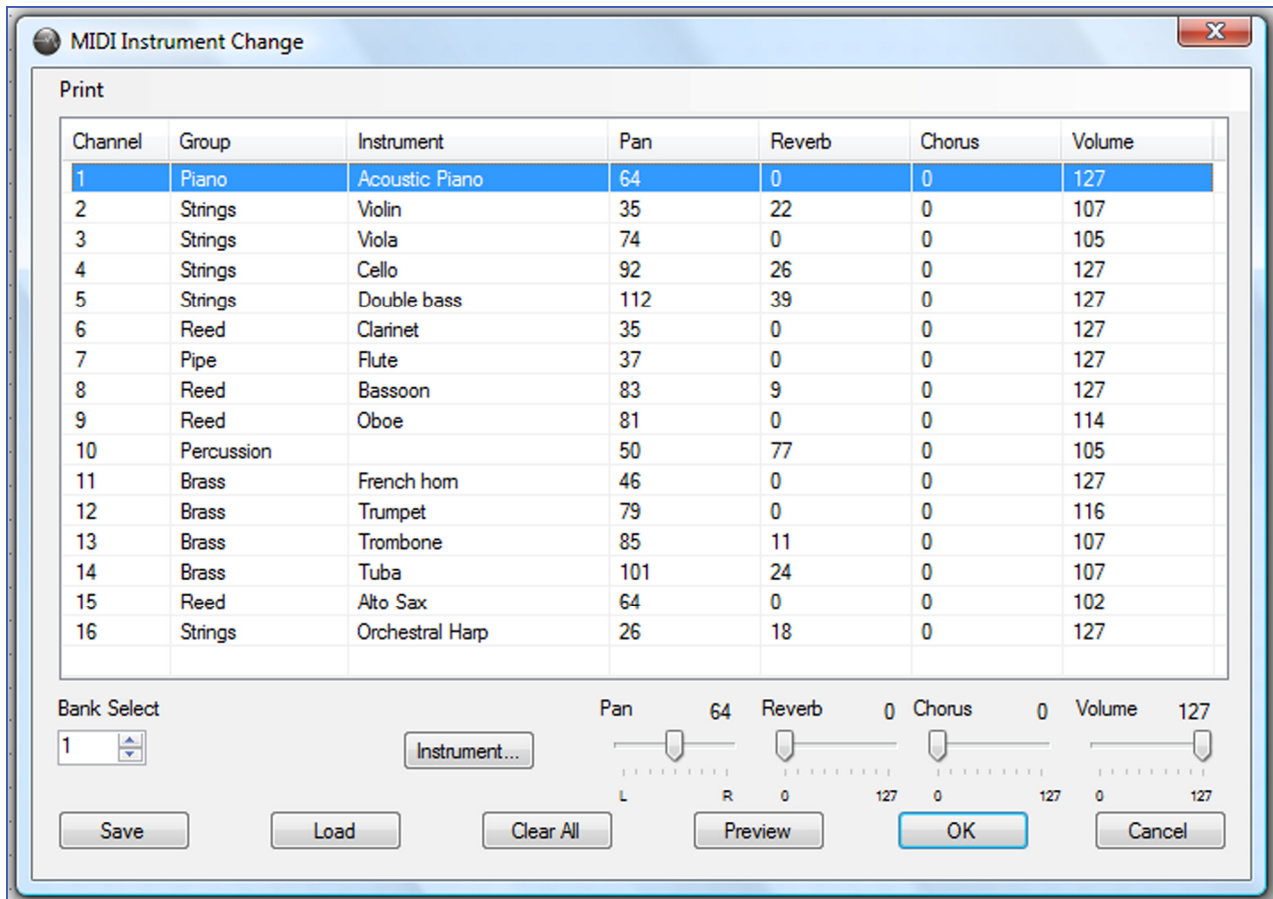
<b>'Trigger'</b> (Switch)	Triggers a change to the new instrument settings. These settings then persist unless any other changes are made by another of these blocks.
---------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

All the settings for this block are made in a separate editing window.

#### Instruments

<b>'Instruments'</b>	The set of MIDI instruments, and values of Volume, Pan, Reverb and Chorus to change for any of the 16 channels. NB. These are not shown in the settings display.
Set using a separate editor window	To open the instruments editing window, double click on the block. Alternatively, you can click on the 'Instruments' setting, then on the down arrow which appears on the right hand side.

## Operating the 'Change Instruments' block editor window:



Task (I want to...)	Operation (how to do it...)
Select channel	In top list, click on a row. Can also then use Cursor up / down keys.
Change instrument for selected channel	<div data-bbox="513 1417 847 1946" data-label="Image"> </div> <p>Click on 'Instrument...' button or double-click on a row in the list.</p> <p>A dialog then opens where you can select instruments from the General MIDI list (see Appendix 3).</p> <p>If you tick the 'Auto Preview' box, you will hear an audition automatically each time you choose a new instrument.</p>

Change 'Pan' setting for selected channel	Drag the 'Pan' slider. 0 = fully left; 64 = centre; 127 = fully right.
Change 'Reverb' setting for selected channel	Drag the 'Reverb' slider. Select a value between 0 (none) and 127.
Change 'Chorus' setting for selected channel	Drag the 'Chorus' slider. Select a value between 0 (none) and 127.
Change 'Volume' setting for selected channel	Drag the 'Volume' slider. Select a value between 0 (silent) and 127. In practice values below about 10 will be too quiet to be heard.
Save settings	Press the 'Save' button, and enter file name
Load settings	Press the 'Load' button, and select file.
Audition the selected instrument / settings	Press the 'Preview' button. NB. This will only happen if you have <i>changed</i> the <i>Instrument</i> .
Cancel all changes	Press the 'Clear All' button.
Exit and save settings	Press the 'OK' button



#### 4.5.5 'MIDI Controller' block



This block receives a variable signal (e.g. from a **Variable Sensor** block) and sends it on the MIDI channel selected. The value received (0-100%) is scaled to create a MIDI value of 0 to 127.

The block has a single input:

<b>'Variable'</b> (Variable)	Change value of the MIDI Controller.
------------------------------	--------------------------------------

#### Controller

<b>'Channel'</b>	The MIDI channel the controller information will be sent on.
1 – 16 (default = 1)	
<b>'Controller'</b>	The controller number to be sent.
0 – 127 (default = 0)	There are 128 different MIDI controllers which can affect the way a sound is produced over MIDI. The controllers are identified by numbers, some of which are associated with specific parameters e.g. 1 is Modulation Wheel.

#### Settings

<b>'MIDI Device'</b> (Int)	The MIDI Output device that the MIDI controller information will be sent to. The specific hardware can be chosen from the 'Ensemble Devices Manager'.
[Audition] [Playback] (default) [Alternate]	



## 4.6 'Video' Palette

The '**Video**' Palette contains a block that can display images and videos on a monitor or external screen.

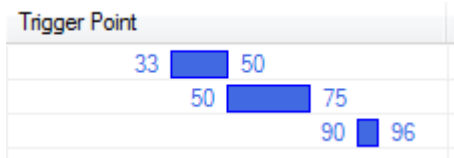
### 4.6.1 'Image and Video' block



Displays images and videos.

It has various inputs that are shown depending on the 'Type' chosen. It can either have a single trigger or up to 6 triggers which control specific images or videos, known as scenes.

**Type:** The block has several different types that allow images and videos to be triggered and controlled in different ways:

[Direct]		Allows up to 6 switches to control separate scenes. Each scene is given its own dedicated trigger input. Several inputs can be triggered at the same time and multiple scenes played. The scene effect can be altered to allow all six to be displayed simultaneously.
[Sequenced]		Has a single input which triggers all scenes in the form of a slideshow. Each scene lasts for a specific amount of time before the next one is shown, and can repeat indefinitely or for a set amount. A 'Scene Duration' value is given, but is overridden by any 'Entry', 'Play' or 'Exit' values for each scene in the 'Effects' tab of the 'Scenes Editor'.
[Stepped]	(default)	A single switch steps sequentially through scenes. Each time you trigger the block, it will show a different scene.
[Variable]		Works in a similar way to using a <b>MIDI Notes</b> block with a <b>Variable Sensor</b> . Each scene is given a range within the input signal (0-100) to which it responds (shown on the screen as blue bars).  The range is set automatically by the block, however this can be adjusted by taking off the 'Auto Trigger' option in the 'Behaviour' tab of the 'Scenes Editor'.
[Numbered]	(Int)	Triggers various scenes depending on the value of the number on the input. The block will display scene 1 when it receives an input of value 1, scene 2 for value 2, scene 3 for value 3 and so on.

The block has a number of settings, divided into categories:



## Block Mode

'Type'		Controls how the scenes play.
[Selective]		Scene to be played depends on which input is active (max. 6).
[Sequenced]		Whilst the trigger is active, scenes will be played similar to a slide show.
[Stepped] (default)		Step to next scene on trigger.
[Variable]		Scene to be played depends on variable signal (e.g. 0-50%, 50-100%).
[Numbered] (Int)		Scene to be played selected by value of trigger (e.g. 1, 2, 3 etc).
'Mode'		Action of trigger input.
[Gate] (default)		Scene is active as long as the trigger is active.
[Trigger]		Scene will continue to its end even if the trigger is removed.
'Allow re-trigger'		Controls what happens if an 'on' switch signal is received whilst the scene or sequence is playing.
[Yes] (default)		The scene or sequence can be "re-triggered" – receiving a trigger ('on' switch signal) will interrupt play and then start the scene or sequence again from the beginning.
[No]		The trigger is ignored – the scene or sequence will keep playing to the end. NB. This setting only matters if the 'Mode' is set to [Trigger]. If 'Mode' is set to [Gate], then the sequence can't be 're-triggered' during playback and this setting will be shown greyed out.

## Image/Video

'Scenes'		Launches the Scenes Editor window.
[Scenes]		Shows when scenes have been created and opens the Editor window.
'Screen'		Choose which screen the Scenes are displayed on.
[Main] (default)		Scenes will be shown on the main screen, for example a laptop screen or main monitor.
[Secondary]		Scenes will be shown on a secondary screen, for example an external monitor or a projector.
'Full Screen'		Shows the video in windowed or in full screen mode.
[No Change]		Leave the screen as it is.
[Windowed]		Shows all scenes on the chosen screen in windowed mode.
[Full Screen] (default)		Shows all scenes on the chosen screen in full screen mode.
'Volume'		Sets the overall volume of the block.
0 – 100 (default = 75)		
'PAN'		Sets the balance between the left and right speakers.
0 – 100 (default = 50)		0 = fully left; 50 = centre; 100 = fully right.
'Priority'		Sets how much priority is given to a particular block if multiple <b>Image and Video</b> blocks are used.
0 – 100 (default = 50)		A block with a high priority (e.g. 100) will display in front of another with a low priority (e.g. 0).



## Sequenced Options

'Loop'		Controls how, or if, the sequence repeats.
[Yes]		The sequence will 'loop', i.e. will repeat playing indefinitely. How the sequence is <i>stopped</i> depends on other settings and connections: (a) If Mode is [Gate], then the sequence will loop while it receives an 'on' signal, and then stop when it receives an 'off' signal. (b) If Mode is [Trigger], then the sequence will continue looping indefinitely - in this case it will only stop when the block's ' <b>Reset</b> ' input is triggered. NB. The 'Repeat Counter' setting (below) is then irrelevant and is shown greyed out.
[No]	(default)	The sequence will not 'loop', however the sequence <i>can</i> still repeat – if 'Repeat Counter' (below) is set to more than 1.
'Repeat Counter'		The number of times the sequence will repeat playing before stopping.
1 – 100	(default = 1)	If 'Mode' is set to [Gate], then an 'off' switch signal will <i>interrupt</i> and stop play before this number is reached. If the 'Loop' setting is [Yes] then the scenes will loop indefinitely, and this 'Repeat Counter' setting is ignored and is shown greyed out.
'Scene Duration'		Default duration for still images.
0 – 100	(default = 5)	Length of time images are displayed in seconds.

## Appendix 1: Glossary

What does it all mean? What is a map? What are signals, sensors, and blocks? Discover it all in this Glossary.

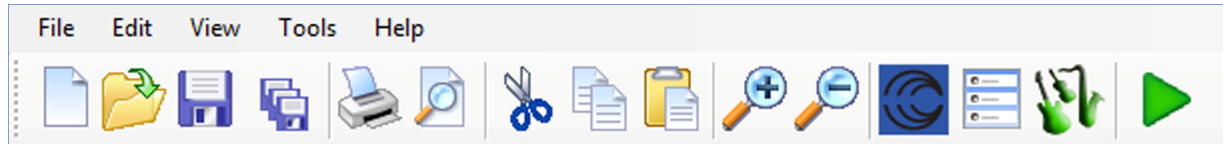
**N.B.** Italics indicate that a term is defined elsewhere within the Glossary

block	<p>An object within your <i>map</i>, shown as an icon on screen.</p> <p>Block names, <b>Switch</b> or <b>MIDI notes</b>, are shown in bold and start with a capital letter.</p> <p>Each block corresponds to various types of activity:</p> <ul style="list-style-type: none"> <li>(i) Inputs - A physical input from sensor or switch;</li> <li>(ii) Tools - Processing or routing of signals;</li> <li>(iii) Outputs - Sound, music, lighting or image output.</li> </ul>
Ensemble Designer	The software which allows you to design a <i>map</i> graphically.
Ensemble interface / Ensemble box	The physical box with sockets into which you plug your <i>switches</i> or <i>sensors</i> . The box communicates with the PC via a USB cable, and transmits signals to <b>Switch</b> or <b>Variable Sensor</b> blocks in <i>Ensemble Designer</i> .
map	<p>The arrangement of connected <i>blocks</i> on screen in <i>Ensemble Designer</i>.</p> <p>The map describes the way <i>signals</i> will be routed and processed (from <i>switches</i> and <i>sensors</i> to sound and light output).</p>
sensor	<p>A physical gadget which you interact with, and which detects variable information like your position or squeeze pressure. An example in the home would be a lighting dimmer switch.</p> <p>Examples are squeeze bulbs, distance or direction detectors (infra red, ultrasonic or magnetic). You can purchase these from Apollo Creative, or other suppliers.</p> <p>When you move with the sensor, this will output a <i>variable signal</i> from the corresponding <b>Variable Sensor</b> block.</p> <p>NB. sensor is written in lower case to distinguish it from a <b>Variable Sensor</b> block.</p>
signal	<p>An item of information traveling between <i>blocks</i> on the <i>map</i>.</p> <p>There are two types of signal: <i>switch signals</i> and <i>variable signals</i>.</p>
switch	<p>A physical switch (similar in action to a light switch), which you press and release. You can buy these from specialists such as QED (<a href="http://www.qedonline.co.uk">http://www.qedonline.co.uk</a>) or Inclusive Technology (<a href="http://www.inclusive.co.uk">http://www.inclusive.co.uk</a>), or even make your own.</p> <p>When you press the switch, this will output an '<i>on</i>' <i>signal</i> from the corresponding <b>Switch</b> block. When you then release the switch, this will output an '<i>off</i>' <i>signal</i> from the <b>Switch</b> block.</p> <p>NB. switch is written in lower case to distinguish it from a <b>Switch</b> block.</p>
<b>Switch</b>	<p>A <i>block</i> which represents a <i>switch</i>. It outputs a <i>switch signal</i> (either '<i>on</i>' or '<i>off</i>'), which is sent to any block(s) to which the <b>Switch</b> block is connected.</p> <p>NB. <b>Switch</b> is written in bold and starts with a capital letter, to distinguish it from a <i>switch</i>.</p>
switch signal	<p>A <i>signal</i> which has two possibilities: either '<i>on</i>' or '<i>off</i>'.</p> <p>A switch signal will often come from a <b>Switch</b> block, when the <i>switch</i> is pressed or release. But <i>other</i> blocks can also output a switch signal, e.g. a <b>Variable to Switch</b> block. This is why we talk about an '<i>on</i>' <i>signal</i>, rather than a "switch press".</p>
trigger	<p>Another way of saying that an '<i>on</i>' <i>signal</i> has been received by a block.</p> <p>This will make the block do something, e.g. play the next note, turn on a light, or output a different value.</p>



'off' signal	<p>A <i>switch signal</i> which is 'off', coming from a switch being released.</p> <p>Similar to 'on' signals, this can trigger an action from other blocks, for example turning off a sound.</p>
'on' signal	<p>A <i>switch signal</i> which is 'on', coming from a switch being pressed.</p> <p>This can trigger an action from other blocks, for example turning on a sound.</p> <p>An 'on' signal <i>could</i> come from a switch being <i>released</i>, if a <b>Switch</b> block has its 'Switch Action' setting set to [Reverse].</p>
<b>Variable Sensor</b>	<p>A <i>block</i> which represents a <i>sensor</i>. It outputs a <i>variable signal</i>, which is sent to any other block(s) to which the <b>Variable Sensor</b> is connected.</p> <p><b>Variable Sensor</b> is written in bold and starts with capital letters, to distinguish it from a <i>sensor</i>.</p>
variable signal	<p>A <i>signal</i> which has a variable value, typically a number between 0 – 100.</p> <p>It can affect other blocks, for example change the volume of a sound.</p> <p>A variable signal will often come from a <b>Variable Sensor</b> block, where the <i>sensor</i> is affected by your movement. But other blocks can also output a variable signal, e.g. a <b>Switch to Variable</b> block.</p>

## Appendix 2: The Menus and Toolbar



The Menus are located above the toolbar.

### File menu

Item	Shortcut	Description
<b>New</b>	Ctrl + N	Creates a new blank map to start designing on.
<b>Open</b>	Ctrl + O	Open an existing map. You can navigate to the appropriate folder and open any .grf file.
<b>Close</b>		Closes just the currently opened map.
<b>Close All</b>		Closes all maps open in Designer.
<b>Save</b>	Ctrl + S	Save the currently opened map.
<b>Save As</b>		Save the currently opened map with a different name.
<b>Save All</b>		Save all the currently opened maps.
<b>Import</b>		Imports a Player file (.grx) to be edited in Designer.
<b>Export</b>		Exports a map so that it can be quickly launched from Player or moved to another computer.
<b>Clear History</b>		Clear the following list of the 10 most recently opened maps.
<b>Print</b>	Ctrl + P	Open the Print window to allow you to print a picture of the currently selected map.
<b>Print Preview</b>		Open a Preview of the currently selected map.
<b>Exit</b>		Closes the Designer software.

### Edit menu

Item	Shortcut	Description
<b>Cut</b>	Ctrl + X	Remove the currently selected block or group and place on the clipboard.
<b>Copy</b>	Ctrl + C	Make a copy of the currently selected block or group and place on the clipboard.
<b>Paste</b>	Ctrl + V	Paste the contents of the clipboard to the current location of the cursor.
<b>Delete</b>	Del	Delete the currently selected block from the map.
<b>Select All</b>	Ctrl + A	Select all the blocks on the currently selected map.

## Align

<b>Tops</b>	Align the top edge of all the currently selected blocks.
<b>Sides</b>	Align the left hand edge of all the currently selected blocks.
<b>Space Equally</b>	
<b>Top To Bottom</b>	Equally space the currently selected blocks between the top and bottom.
<b>Left To Right</b>	Equally space the currently selected blocks from left to right.

## View menu

Item	Shortcut	Description
<b>Zoom In</b>	Ctrl +	Zoom in closer to the current map.
<b>Zoom Out</b>	Ctrl -	Zoom further out from the current map.
<b>Zoom 100%</b>		Go back to the original zoom level.
<b>Zoom To Fit</b>		Automatically scale the zoom to fit all the blocks on the currently selected map.
<b>Show Grid</b>		Show the grid behind the blocks on the map.
<b>Snap To Grid</b>		When moved on the map, all blocks will automatically move to the nearest grid point.
<b>Reset Size</b>		Reset the size of a map that has been enlarged by scrolling.

## Tools menu











Item	Shortcut	Description
<b>Play</b>	F5	Play the currently opened map in the Player application.
<b>Stop</b>	Shift + F5	Stop the currently playing map.
<b>License</b>		Change the license registered by the software to enable new features.
<b>Ensemble Device Monitor</b>		Open the Ensemble Device Monitor to view and make changes to the connected hardware.
<b>Palette Manager</b>		Open Palette Manager to change the blocks that appear in the palette.
<b>MIDI Instruments</b>		Change the default set of 16 MIDI Instruments associated with the currently selected map.

## Help menu

Item	Shortcut	Description
<b>About</b>		Show the software version number and copyright information.
<b>Apollo Ensemble Manual</b>		Open the Apollo Ensemble manual in pdf format.
<b>Help</b>	F1	Launch the help file.



The Toolbar contains shortcuts to commonly used features in the software.

Icon	Name	Description
	<b>Create a new map</b>	Creates a new blank map to start designing on.
	<b>Open an existing map</b>	Open an existing map. You can navigate to the appropriate folder and open any .grf file.
	<b>Save this map</b>	Save the currently opened map.
	<b>Save all maps</b>	Save all the currently opened maps.
	<b>Print this map</b>	Open the Print window to allow you to print a picture of the currently opened map.
	<b>Print preview</b>	Open a Print Preview of the currently selected map.
	<b>Cut to clipboard</b>	Remove the currently selected block or group and place on the clipboard.
	<b>Copy to clipboard</b>	Make a copy of the currently selected block or group and place on the clipboard.
	<b>Paste from clipboard</b>	Paste the contents of the clipboard to the current location of the cursor.
	<b>Zoom in</b>	Zoom in closer to the current map.
	<b>Zoom out</b>	Zoom further out from the current map.
	<b>View connected devices</b>	Open the Ensemble Device Monitor.
	<b>Palette Manager</b>	Launch the Palette Manager to add, remove and change blocks.
	<b>MIDI Instruments for this map</b>	Open the MIDI Instruments selector to change which instruments are assigned to which MIDI channel.
	<b>Play/Stop</b>	Play/Stop the currently open map in the Apollo Player.

## Appendix 3: What is MIDI?

MIDI is a standard way of communicating musical information between electronic musical instruments – any MIDI keyboard can play sounds on any MIDI synthesiser or be recorded on any computer. For example, we can tell a MIDI instrument to play a particular note pitch, on a particular instrument, and how loud to play it.

MIDI does this using digital codes transmitted along a MIDI cable (5 pin DIN) or USB cable. Ensemble Designer uses MIDI codes to define what notes and chords to send, so it is useful to know a few details about this:

- Values in MIDI range from 0 to 127, giving us 128 possible notes (pitches).
- The loudness/volume of each note is set by a 'velocity' value from 0-127 (the name derives from the 'speed' at which a key on a keyboard would have been hit).
- To play more than one instrument at the same time, MIDI uses 'channels' numbered 1 – 16. Each channel corresponds to a separate instrument that can play independently of the others, just like a small orchestra.
- The sound in use on each channel can be changed at any time by sending a command to select one of 128 different instruments (**Change Instruments** block).

'General MIDI' (GM) is an extension of MIDI which, amongst other things, defines the 128 instruments and reserves channel 10 for percussion. This enables people to know their music will sound similar, even if played on different systems.

An analogy may help make this clearer. Imagine that a conductor is working with a group of sixteen musicians, each of whom has a pile of instruments at their feet. The conductor can tell any musician to play different notes, and also tell them to pick up a different instrument. GM means that all conductors and musicians have the same pile of instruments at their feet, with the same names. So any conductor can go and work with any group of musicians, and have a fair chance of creating a similar-sounding performance.

You do need to bear in mind that the quality of your sounds will vary according to the quality of your equipment (sound card, PC etc.), in the same way that a professional orchestra with expensive instruments will sound better than a school orchestra using cheap ones.

## Appendix 4: What is DMX?

DMX is a standard way of controlling lighting equipment (similar to the way MIDI controls music), for example to turn lights on or off, change colours or positions. Each DMX cable can control up to 512 different lighting parameters (called 'channels') at the same time, with values ranging from 0 – 255. Cables can also be 'daisy-chained' between pieces of equipment, which greatly cuts down the amount of cabling.

Each light fixture has a number of channels defined by the manufacturer. For example, a moving-head light might respond to three DMX channels - one controls the 'pan', one the 'tilt' and one the 'brightness'. To distinguish which light fixture we want to control, each has a DMX 'base address'. For example, if the 'base address' for this light is set to 32, then its 'pan' will be controlled on DMX channel 32, its 'tilt' on channel 33 and its 'brightness' on channel 34.

Now, if we had a second light of the same type, we would set its 'base address' to 36, and this light would then be controlled by sending commands on DMX channels 36, 37 and 38.

As well as lights, DMX can also control other equipment such as smoke machines or special effects. By using a DMX dimmer or switch pack, it is also possible to use DMX to send commands to virtually any mains operated effect or light.

## Appendix 5: Declarations of Conformity

### Declaration of Conformity

**Manufacturer's Name** GSPK Design Ltd

**Manufacturer's Address** Manse Lane  
Knaresborough, North Yorkshire  
HG5 8LF, England

**declares, that the product**

**Product Name:** Ensemble 16  
**Model Number:** D830203

**Conforms to the following Product Specifications:**

#### **Safety**

**BS EN 60335-1 : 2002**

Household and similar electrical appliances – Safety – Part1: General Requirements.

#### **EMC**

**BS EN55022 : 1998**

Information Technology Equipment – Radio Disturbance Characteristics – Limits and Methods of Measurement.

**BS EN55024 :1998**

Information Technology Equipment – Immunity Characteristics – Limits and Methods of Measurement.

#### **Supplementary Information:**

The product herewith complies with the requirements of the EMC Directive 2004/108/EC and carries the “CE” mark accordingly.



Paul Marsh  
Managing Director  
Knaresborough  
20<sup>th</sup> July 2009

## Declaration of Conformity

**Manufacturer's Name** GSPK Design Ltd

**Manufacturer's Address** Manse Lane  
Knaresborough, North Yorkshire  
HG5 8LF, England

**declares, that the product**

**Product Name:** **Apollo Creative Wireless System**

**Model Number:** E110104 (Ensemble Hub)  
E110206 (Ensemble Connect)  
E110208 (Ensemble Dice)  
E110304 (Ensemble Dual)  
E110306 (Ensemble Tilt)  
E110308 (Ensemble Press)  
E110317 (Ensemble Tag)

**Conforms to the following Product Specifications:**

### **Safety**

**BS EN 60335-1 : 2002**

Household and similar electrical appliances – Safety – Part1: General Requirements.

### **EMC**

**BS EN 55022 : 1998**

Information Technology Equipment – Radio Disturbance Characteristics – Limits and Methods of Measurement.

**BS EN 55024 :1998**

Information Technology Equipment – Immunity Characteristics – Limits and Methods of Measurement.

### **Supplementary Information:**

The product herewith complies with the requirements of the EMC Directive 2004/108/EC and carries the “CE” mark accordingly.



Paul Marsh  
Managing Director  
Knaresborough  
18 February 2010