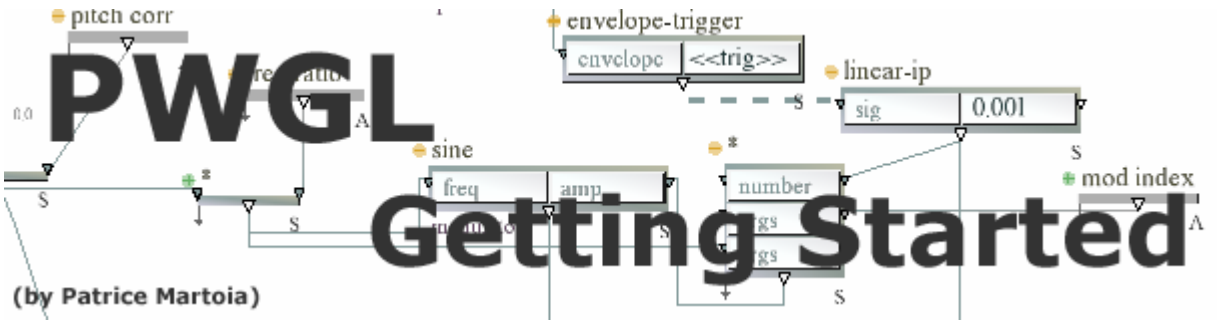


Introduction

PWGL is a modern graphical programming environment for musicians developed by Mikael Laurson, Mika Kuuskankare, and Vesa Norilo from the Sibelius Academy in Finland.

Like IRCAM's OpenMusic, PWGL is a direct descendant of PatchWork and excels in the fields of computer-assisted composition and music analysis; in particular, it offers a notational front-end and a general purpose constraint-based language with a strong emphasis in music related search problems.
However, unlike OpenMusic and rather like Cycling74's Max/MSP, PWGL is also a flexible real-time sound-synthesis system.

The purpose of this short document is to help newcomers to PGWL to get started as painlessly as possible.

Setup

Installation

Although PWGL is not open-source software, it can be freely downloaded from http://www.siba.fi/PWGL/downloads.html; the platforms currently supported are Mac OS X and Windows XP.

PWGL is available in two flavours:

- The PWGL Application, a stand-alone application; and
- The PWGL Binaries, a set of pre-compiled Lisp files that require LispWorks.

Please note this document assumes the reader chooses the PWGL Application package for Windows XP.

The installation of the software itself is very simple for Mac OS X 10.4 and Windows XP as an installer is provided.
The procedure for Mac OS X 10.3 is a little more complicated and is clearly detailed on the webpage.

Once you have installed the application, start it up to check that the process has been successful; three windows should appear, as shown below:

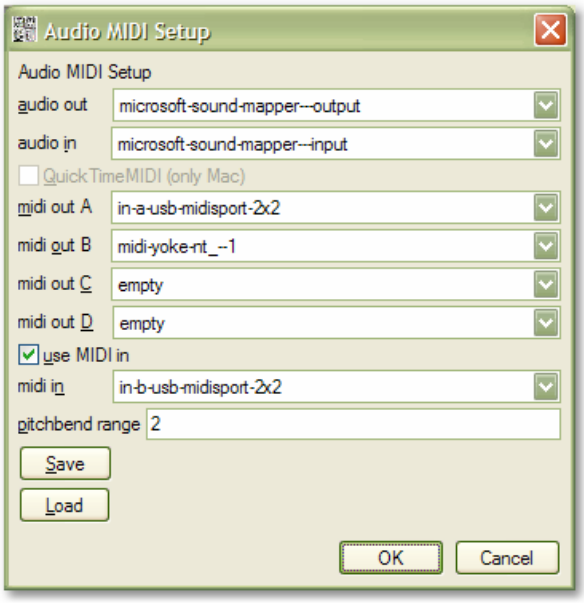
1. The Listener window (*PWGL 1.0*);
2. The Output Browser window (*PWGL output*); and
3. The Patch window (*PWGL1*).

The nature and purpose of these windows will be discussed later in the document.

Audio and MIDI setup

The *PGWL* menu from the workspace window lets you set up your preferences, including the Audio and MIDI settings.

If you choose the *Audio MIDI Setup Preferences* option (*PGWL|Audio MIDI Setup Preferences*), the window below will appear on the screen. Select the audio and MIDI devices to be used and save your selection.



PWGL lets you choose up to four MIDI-out ports (channels 1-16 for the device *A*, channels 17-32 for *B*, channels 33-48 for *C*, and channels 49-64 for *D*), and one MIDI-in port.

Please note the *pitchbend range* value should match the pitch-bend settings of your MIDI synthesizer. This parameter is used for micro-tonal tuning purposes.

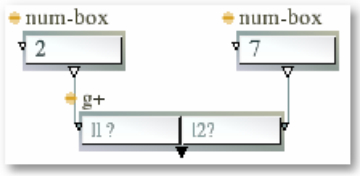
Overview

This section describes the PWGL application and introduces basic terms and concepts that will be used throughout the rest of the document.

Visual programming with PWGL

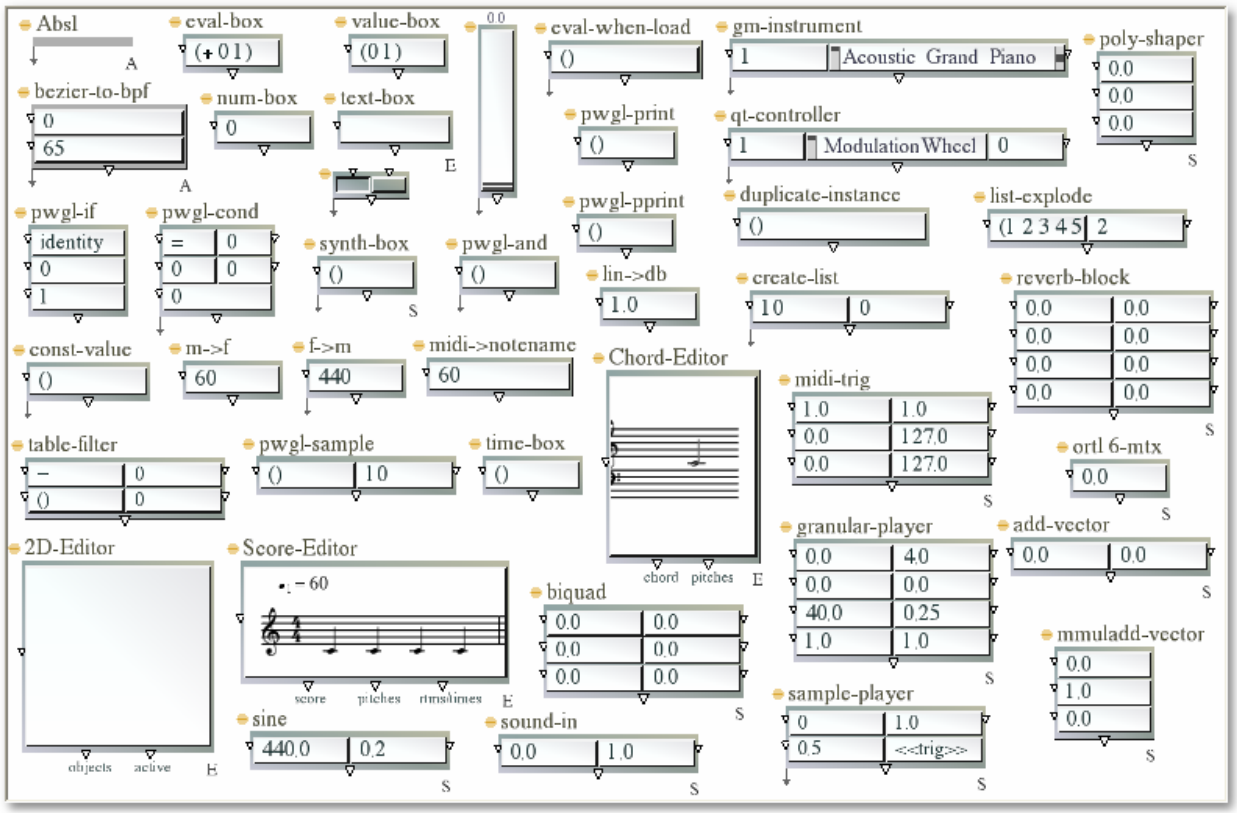
Programs in PWGL are patches that contain interconnected objects.
On the screen, these objects appear as boxes that contain usually some text or numbers, but complex objects often require more complex representations. It is by connecting these boxes together that you write PWGL programs.

For example, this very simple patch calculates the sum of two numbers.



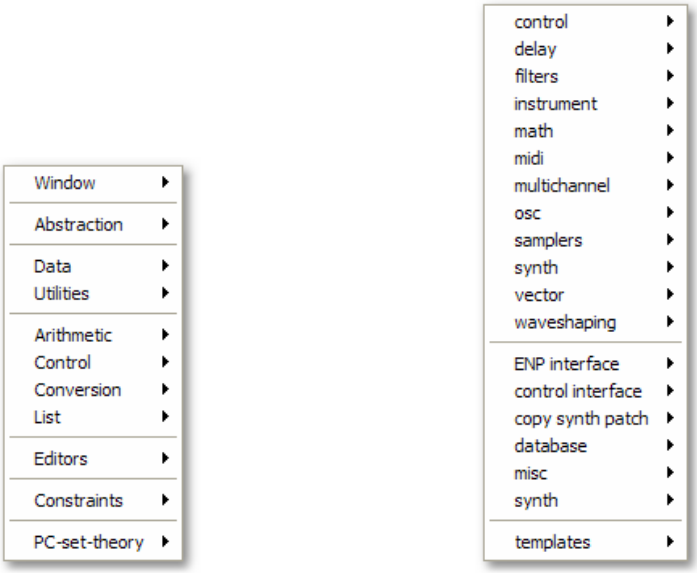
Objects and boxes

PWGL has several dozens of built-in objects available, categorised by type (e.g., Arithmetic or Wave shaping): the picture below shows a random selection.



Before you can select an object, you first need to navigate to its category by either

- Clicking the mouse right button (right-click), or
- Pressing the *Shift* key whilst clicking the mouse right button (*Shift* + right-click).



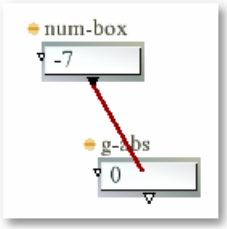
This displays one of the popup menus shown above; you can then choose the appropriate category sub-menu and select the object you want.

It is also worth remembering that most objects have properties you set by double-clicking on their box.

Windows

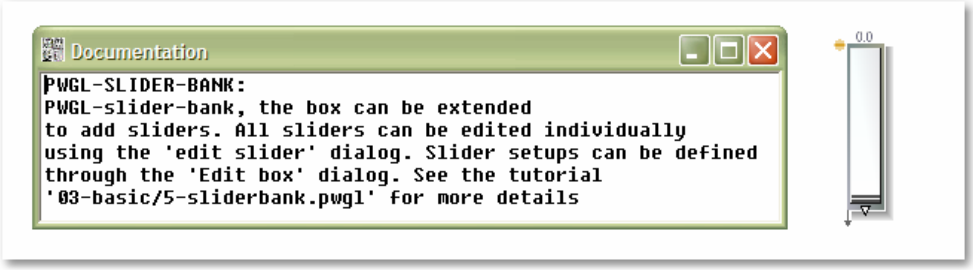
Programs are created inside ***patch windows*** similar to Max/MSP's Patchers. PWGL always displays at start-up an empty patch window, *PWGL1*.

As explained earlier, programs are written by connecting objects together. To do so, click on one of the outlets of one object (one of the little triangles located right under it) and drag the patch cord onto the other object. When dragging the cord inside the target box, it will become red to indicate that you can release the mouse button to make the connection.



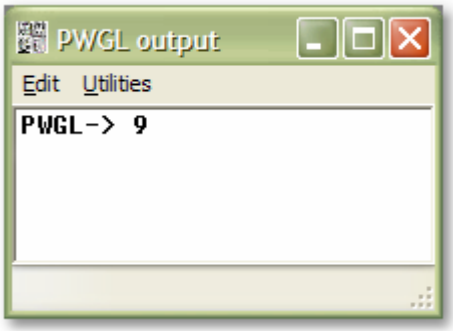
You can perform all the usual editing operations on the object boxes, such as copy (*Ctrl+C* or *Edit|Copy*), paste (*Ctrl+V* or *Edit|Paste*), cut (*Ctrl+X* or *Edit|Cut*), move them with your mouse, etc.
To select a box, click on the text just above it; note that the cursor changes into a hand and the text turns red after you have clicked on it.

PWGL displays information on patches or individual objects when you press the key ***d*** (for *d*ocumentation) on your keyboard.
If no objects are selected, a window pops up with a descriptive text on the patch being edited inside. If one or more objects are selected, the window contains the relevant documentation instead.

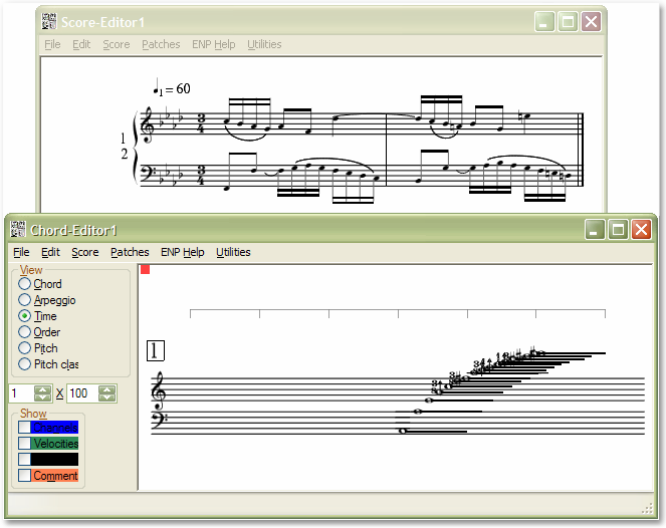


Besides *d*, a few other keys have special functions in patch windows when applied to specific objects.

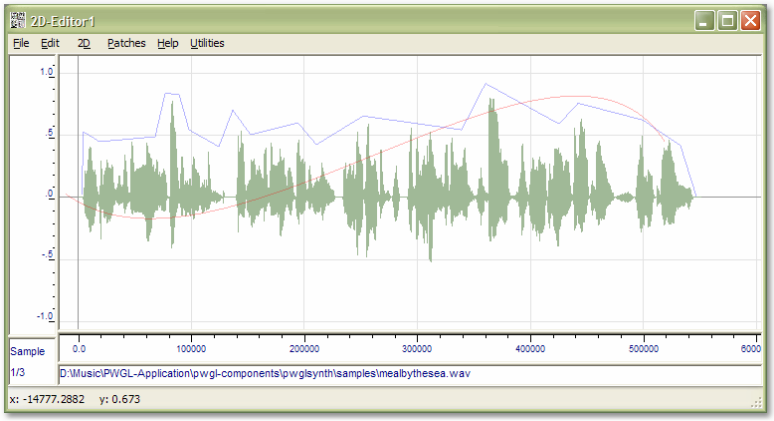
The letter ***v*** is used to e*v*aluate a patch.
For example, the patch shown earlier that adds up two numbers does not directly display the result: to evaluate it, select the *g+* object and press *v*; the value 9 appears in the ***PWGL output window***. Whenever you evaluate a patch, the output window will display the resulting value.



Press ***p*** to *p*lay (MIDI) chords or scores in *Chord-Editor* and *Score-Editor* objects respectively.



The ***space*** key plays samples in *2D-Editor* objects.

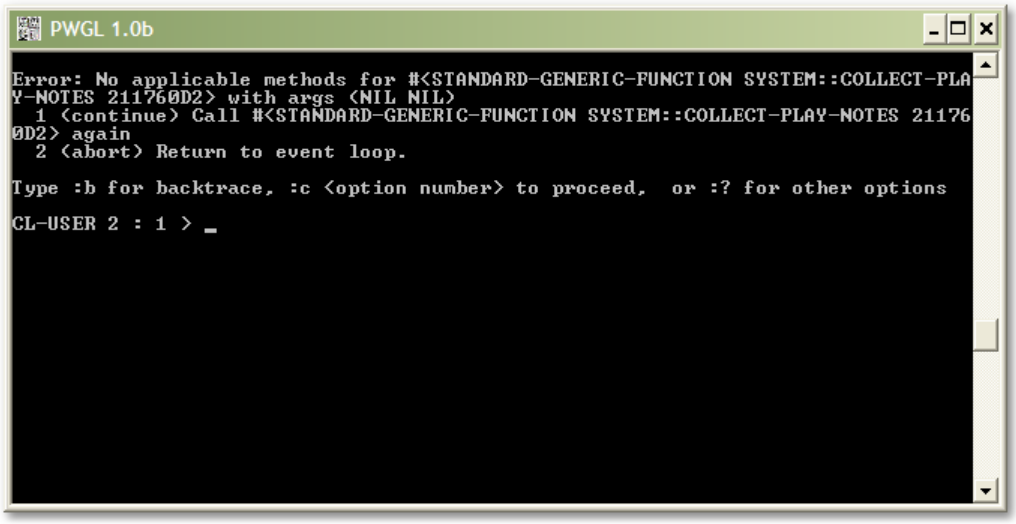


Finally, the ***s*** key is used to stop the audio playing.

When you have finished composing your program, you can save it (*Ctrl+S* or *File|Save*, or *File|Save As…*) as a *.PWGL* file; you will able to re-open it later (*Ctrl+O* or *File|Open…*).

At times, especially when you try and evaluate patches, PWGL seems to freeze. If you persist and attempt to perform any operation in the patch window, the dreaded *(Not Responding)* message may even appear in the title bar.

Even though it may seem the application has crashed (and you have lost all your changes since the last time you saved the patch), it probably has not and is waiting for your decision about how to handle some unexpected problem in the ***listener window***.



There, you are presented with the description of the error and several options to track down the cause of the problem (enter ***:?*** to see the full list); if you are merely interested in resuming the session, enter ***:a***.

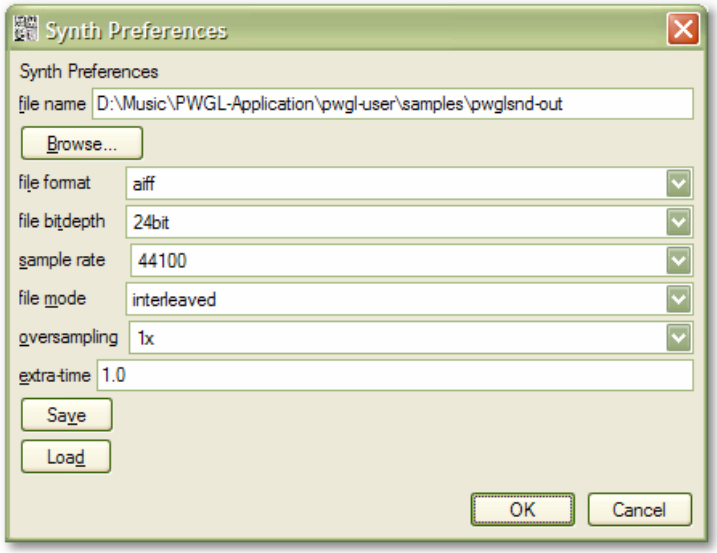
Sound synthesis

This section illustrates with a very simple example how real-time sound synthesis can be achieved with PWGL.

PWGLSynth

PWGL, unlike its ancestor PatchWork, does offer real-time sound synthesis via a dedicated component, *PWGLSynth*.

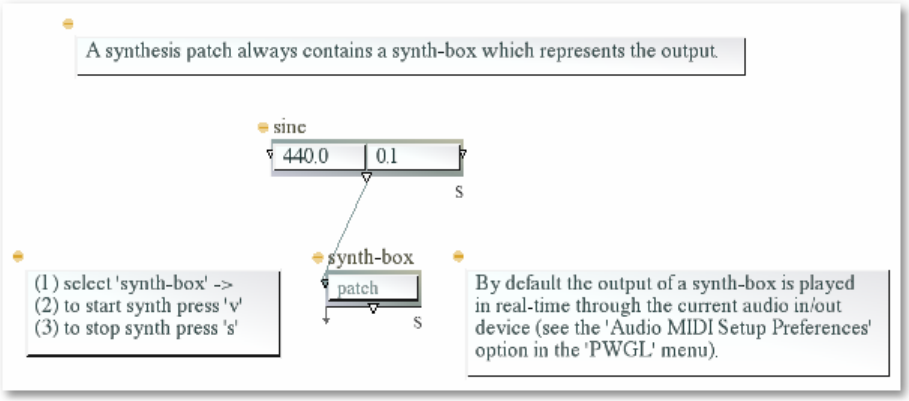
PWGLSynth can be configured through *PWGL|Synth Preferences…*.



Patch-wise, there is a graphical object called a *synth-box* that allows you hear the sounds you generate within the patch window.

A simple patch

Some of the tutorials packaged with PWGL provide useful in-depth examples of sound synthesis orientated patches.
An excellent starting point is the *08-Synth/02-Basic/01-sine.pwgl* tutorial.



To hear the sine-wave, *ev*aluate the synth-box (i.e., select the object and press *v*). Conversely, press *s* to *s*top the audio.

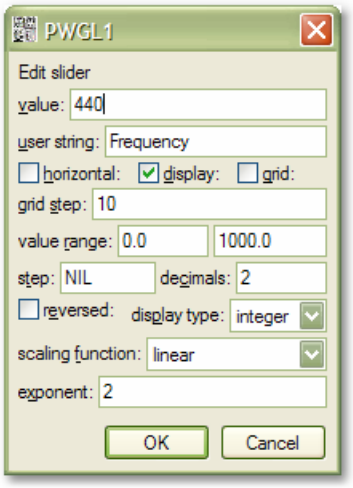
You can access the documentation available on the sine object by selecting it and then pressing *d*. It reads *SINE (FREQ AMP)* which means it takes two parameters: the frequency on the left inlet and the amplitude on the right one.

You can change either of these values if you click on it and, while keeping the mouse button down, move the mouse up and/or down.
Please note that if you do so while the synth-box is being evaluated, you can hear immediately the effect.

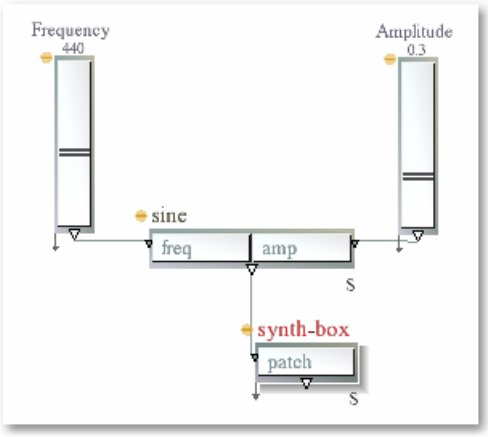
In order to make the patch easier to use, you can add sliders (*bank-slider* objects) to better control the frequency and amplitude.

Connect one of the sliders to the frequency inlet of the oscillator and the other to its amplitude inlet. Note that the values 440.0 and 0.1 have been replaced in the sine box by *freq* and *amp*; this indicates that the sliders are now the main means to control the parameters.

If you double-click on the slider you have connected to the frequency inlet of the oscillator, you will be able to give the slider a label (*user string*), and set its default *value* and its *range*.



Assuming you did the same with the amplitude slider, the modified patch should look like this:

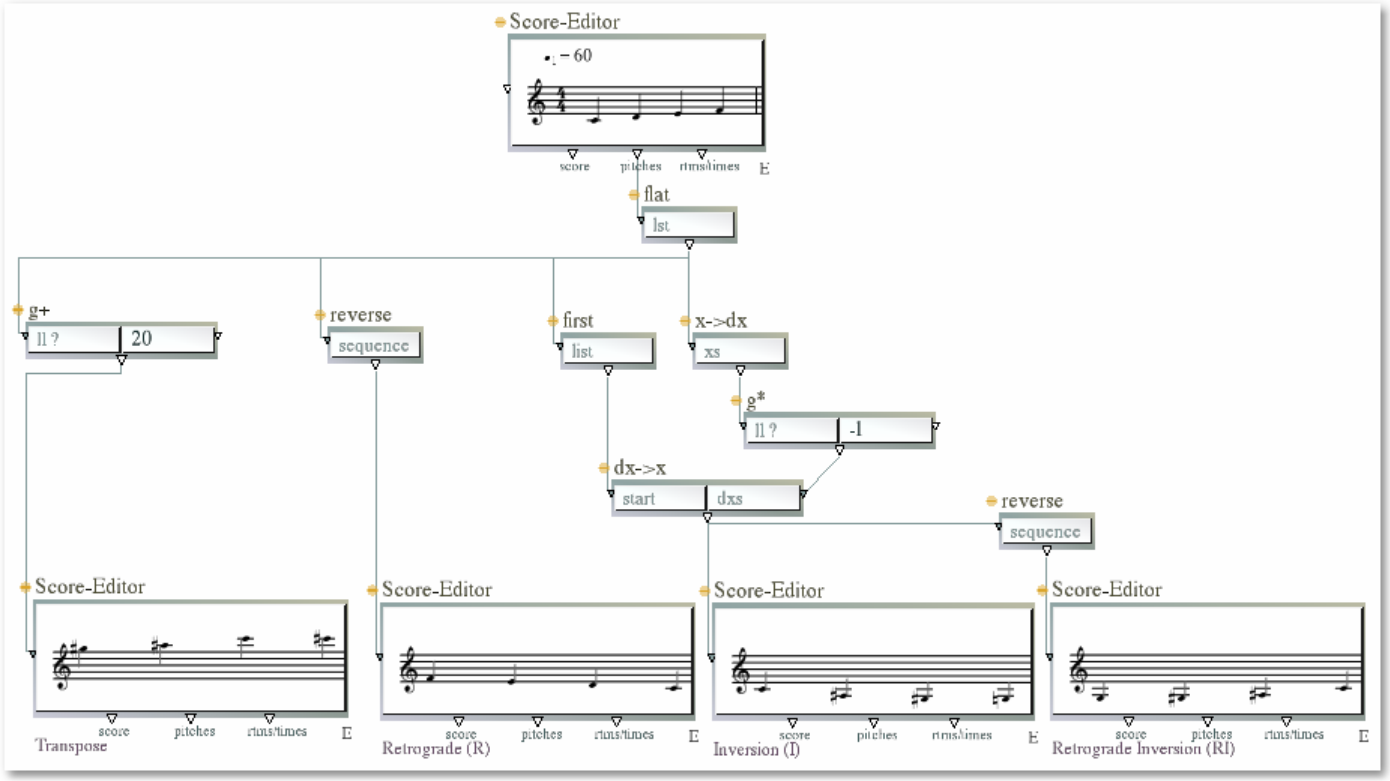


Computer-assisted composition

Both PWGL and OpenMusic are based upon PatchWork. It is therefore hardly suprising that both offer a similar set of functionality in the area of computer-aided composition.

A not-so-simple patch

The patch below calculates the derived forms (transposed, retrograde, inverted, and retrograde of inversion) of a list of pitches.



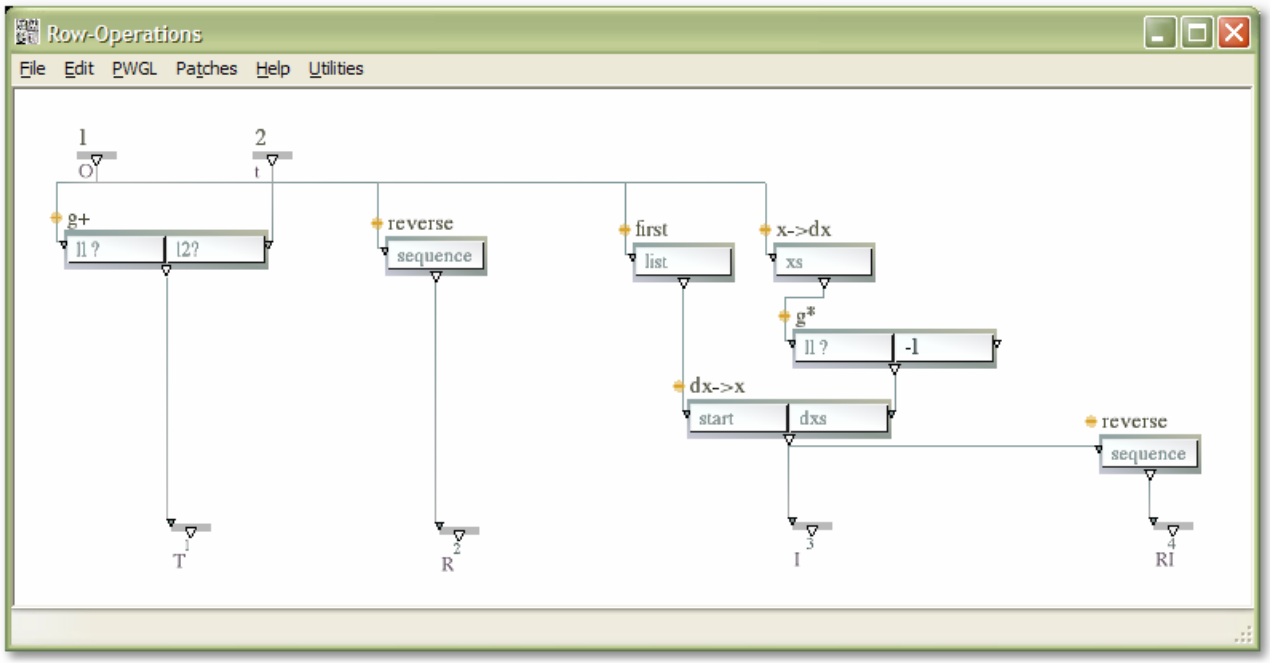
To check the result for each operation, e*v*aluate the relevant Score-Editor (i.e., select the object and press *v*).

Let us go quickly through the patch.
The Score-Editor object at the top contains the basic set. Score-Editors can output their content under different formats and it makes sense to use the pitch list here; in the example above the Score-Editor evaluates to *((((((60)) ((62)) ((64)) ((65)))))))*. The function *flat* from the package *pw* is used to obtain the simpler, "flatter" list *(60 62 64 65)*.
The transposed set is easily calculated by adding a numerical value to each element in this list. Add *20* as above and you obtain *(80 82 84 85)*.
The retrograde set is simply the list reversed. The Lisp function *reverse* returns *(65 64 62 60)*.
The inverted set is a little bit more complicated to derive. Intervals between the elements taken two by two are calculated by *x->dx* from the package *pw,* before being multiplied by *-1* to obtain the list *(-2 -2 -1),* which is then used along with the first element of the original list by *dx->x* to obtain the inverted set, *(60 58 56 55)*.
Finally, the retrograde of inversion is the inverted set reversed, *(55 56 58 60)*.

As it is likely that these operations will be performed in future compositions on various lists of pitches, it is worth encapsulating them into an ***abstraction*** (a type of reusable component that abstracts away its internal structure thus simplifying complex patch layouts).

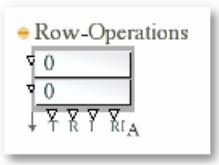
Generic (empty) *abstract-box*es have no inputs or outputs; these can be created from inside the abstraction patch window (double click on the abstract-box to open its patch window, then right-click anywhere in the patch window and choose *Abstraction*|*abstract-input* or *abstract-output*).

Create an empty abstraction, open it and create or copy over the objects that describe the logic behind the operations above. You will need to create two inputs (one for the pitch list and one for the transposition value. Four outputs are also needed for the transposed, retrograde, inverted, and retrograde of inversion pitch lists. Finally, set the name of the abstraction to *Row-Operations*.
The resulting patch should be close to the following:

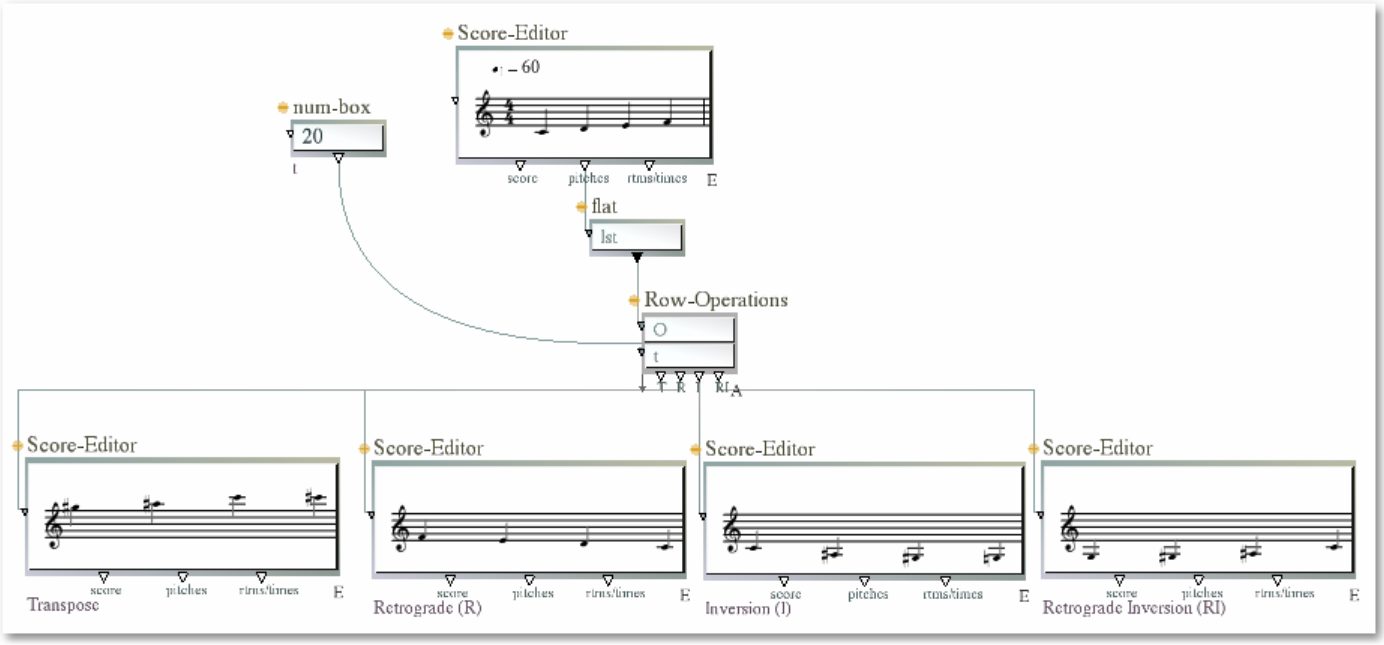


Save the abstraction (either close its patch window or select its box, right-click and choose *save abstraction*).
Note that PWGL stores the abstractions you create in the folder *C:\Documents and Settings\<user name>\PWGL-User\Abstractions*.

The Row-Operations box should look like this, with two inlets and four outlets to match the inputs (*O* and *t)* and outputs (*T*, *R*, *I*, *RI*) in the abstraction patch.



The next and final step is to rewrite the original patch as shown below.


Score-Editor
num-box
20
score
pitches
rtms/times
E
flat
lst
Row-Operations
O
t
T
R
I
RI
A
Score-Editor
Score-Editor
Score-Editor
Score-Editor
Transpose
Retrograde (R)
Inversion (I)
Retrograde Inversion (RI)

Appendix A – Keyboard shortcuts

The full list of keyboard shortcuts can be inspected through *Help*|*Keyboard Shortcuts*. The table below is a selection of the most frequently used ones in patch windows.

Ctrl + .	Stop PWGL processes
Ctrl + A	Select all objects in the window
Ctrl + C	Copy the selected object(s)
Ctrl + D	Duplicate the selected object(s)
Ctrl + E	Export (e.g., as EPS or as a MIDI file)
Ctrl + F	Make the patch fit in the window
Ctrl + I	Load the instrument library
Ctrl + N	Open a new patch
Ctrl + O	Open a patch
Ctrl+ Q	Exit PWGL
Ctrl + S	Save the current patch
Ctrl + T	Open the PWGL tutorial window
Ctrl + U	Load a PWGL or User library
Ctrl + V	Paste the object(s) from the clipboard
Ctrl + W	Close the window
Ctrl + X	Cut the selected object(s)
Ctrl + Z	Undo the last operation

Appendix B - Resources

Related to PWGL

- PWGL (http://www.siba.fi/PWGL/index.html)
- OpenMusic (http://freesoftware.ircam.fr/rubrique.php3?id_rubrique=15)
- LispWorks (http://www.lispworks.com/)

Other audio software

- Max/MSP (http://www.cycling74.com/products/maxmsp)
- Pure data (http://puredata.org/)
- Reaktor (http://www.native-instruments.com/index.php?id=reaktor5_us)