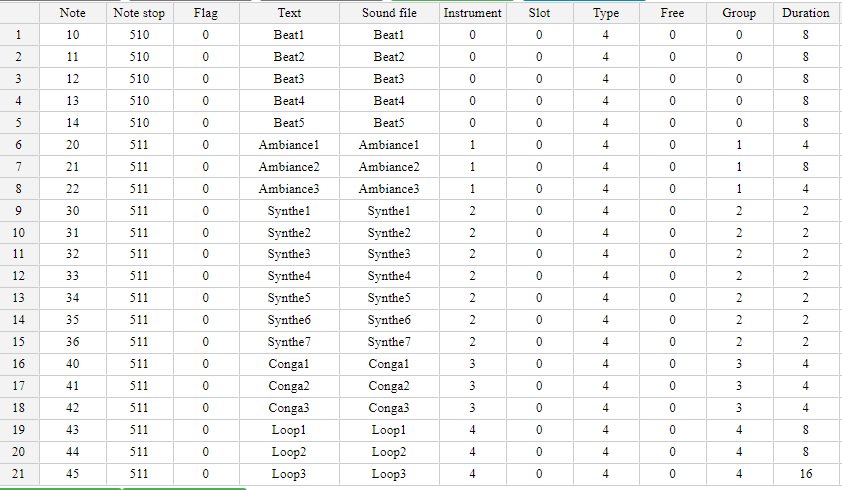
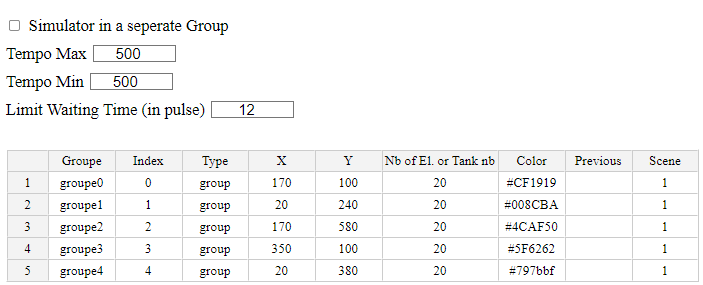
Here are the patterns descriptors and the groups we will use in our tutorials.





# A first method for Setting Groups

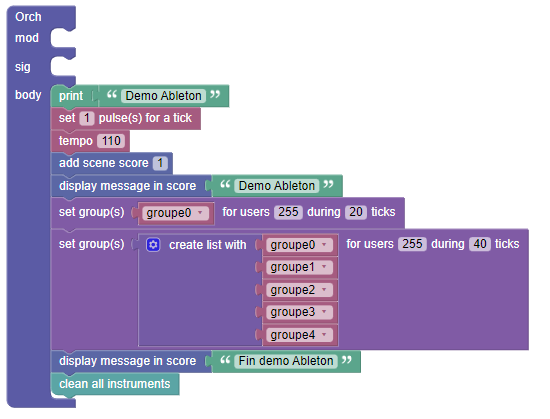
There are several ways of presenting (or activating) groups of patterns to the simulator or the audience. This example shows how to present groups during a period. Skini receive “pulses” according to the synchronization mechanism chosen. We could decide to set the “tick” according to a multiple of these pulses. The impact of the choice will be explained later. We chose to have a tick per pulse.

“add scence score” will allow us to see the result of our program on the “score” window. It is set to 1 according to what we put in the group description where the “scene” of the groups is set to 1.

We first set the group 0 with a time limit of 20 ticks.

Then we set several groups together for 40 ticks.

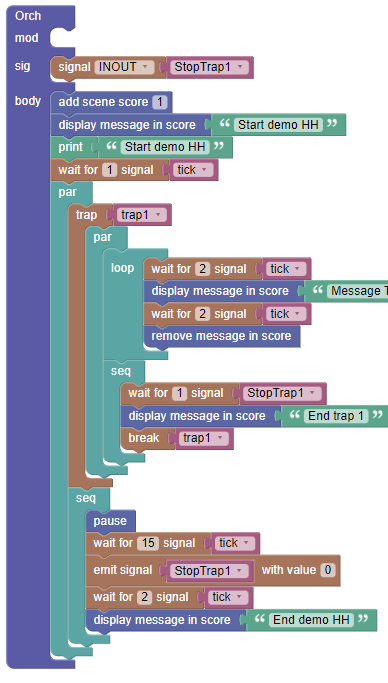
“clean all instruments” is a way to empty the FIFO of the instruments which have been filled by the simulator.



# Sequence and Parallel

# Loop, every, loopeach

# Manage Traps

In this example we will create and use a signal: StopTrap1. (demoHHwait.xml).

We have first to declare the signal in the “sig” block. The signal is INOUT. It can be used for emitting and receiving. Most of the time we can use this option set to INOUT.

“add scence score” will allow us to see the result of our program on the “score” window. It is set to 1 according to what we put in the group description where the “scene” of the groups is set to 1.

“print” is a way to get information on the console.

We have several “wait for” blocks in this example. They allow to stop the program until a specific signal arrives. Signals must be first created as variables to be used.

You already know about “seq” and “par”.

The goal here is to stop the “loop” when a “StopTap1” signal is emitted. For that we have the trap in parallel with a simple way of emitting the “StopTrack1” signal using some “ticks”. The “break” block kill the trap.

# Wait and emit signals

