MANUAL DE INSTRUÇÕES

**Vendaval** é um algoritmo de síntese sonora escrito em Csound pela framework do Cabbage desenvolvido por Caio M. Jiacomini.

Se houver qualquer dúvida sobre seu funcionamento após a leitura, sinta-se livre para entrar em contato comigo por email (caiojmini@gmail.com). O repositório no Github com o código fonte pode ser acessado nesse link: https://github.com/CaioMJ/Vendaval

**Vendaval** é composto por seis componentes:

**GLOBAL:**

Controla atributos que governam o plugin como um todo.

É importante notar que os valores do envelope ADSR devem ser definidos antes a nota MIDI ative o instrumento. Se você mudar o valor do decay, sustain, ou release após o instrumento ser ativado, os novos valores não serão aplicados, portanto, **você deve definir os valores do envelope do jeito que você desejar antes de ativar o instrumento com uma nota MIDI.**

**WOOING:**

Produz ruído filtrado para simular o assovio do vento. Ruído rosa é filtrado com um bandpass com a frequência central sendo modulada para valores aleatórios em tempos aleatórios.

**BACKGROUND:**

Funciona quase idêntico ao componente de wooing, exceto que o alcance de frequência é mais limitado e a largura de banda é maior.

**GUSTS:**

Também funciona de modo similar aos componentes de wooing e background, a diferença principal sendo que a frequência de centro do bandpass é modulada também por um LFO, criando um som mais ondulante.

**RUMBLE:**

Produz ruído com um low-pass filter. Ao invés da frequência de centro de um bandpass ser ajustável, há um parâmetro para ajustar a frequência de cutoff do low-pass, que não é modulada aleatoriamente como os outros filtros.

**REVERB:**

Aplica reverb para todos os outros componentes.

**PARÂMETROS:**

**Volume:** controla o volume dos componentes individualmente

**Frequency/Cutoff:** controla a frequência central do filtro de bandpass primário ou a frequência de corte do low pass

**Range**: controla o alcance de frequência que o bandpass pode ser modulado, tanto acima quanto abaixo da frequência central. Age como um multiplicador para um número definido no código

**Rate:** controla a frequência com a qual o bandpass é modulado. Age como um multiplicador para uma extensão de valores de tempo

**Bandwidth** controla a largura de banda do bandpass

**Distortion:** controla a distorção aplicada ao componente de Rumble

**Harmonizer Vol:** controla o volume de um filtro bandpass secundário que harmoniza com o primário. O filtro secundário é usado para similar o som resonante criado quando o vento assopra por uma passage estreita.

**Harmonizer Freq:** multiplica a frequência do filtro primário pelo valor definido nesse parâmetro para definir a frequência do filtro secundário. Resultados mais realistas podem ser encontrados mantendo o valor entre 1.1 e 2

**Attack, Decay, Sustain, Release:** define o envelope de amplitude geral do instrumento

**Mix:** define o equilíbrio entre o sinal não afetado e o sinal afetado pelo

reverb

**Size:** define o tamanho do reverb

**PRESETS**

Clique no botão “Save” para salvar a configuração atual como um preset

**GERENCIAMENTO DE PRESETS**

Renomear e organizer presets é um pouco complicado com a framework que eu usei para desenvolver o **Vendaval**, então aqui vai minha tentativa de explicar como funciona.

As configuracões de presets são guardads dentro de um arquivo de texto de formato *.snaps*

No Windows, esse arquivo fic ana mesma pasta que o plugin.

No MacOS, esse arquivo pode ser acessado clicando com o botão direito no arquivo do plugin e selecionando “Mostrar conteúdo do pacote”. O arquivo *.snaps* fica dentro da pasta de “Contents”

Aqui vai uma imagem do arquivo de presets:

Text

Description automatically generated

O que esse arquivo faz é: definer a ordem dos presets, definir o nome do preset e guardar os valores de cada parâmetro para o preset

Cada preset começa com **<PRESET#** e termina com **/>**

Eu destaquei abaixo onde começa e termina o **PRESET0** para ficar mais claro:

Text

Description automatically generated

Após definir o número do preset (PRESET, PRESET1, PRESET2), o texto define o nome do preset com **PresetName=“name”**. O primeiro preset se chama “Reset” já que é declarado como **PresetName=“Reset”**

Para mudar o nome, basta substituir o texto que estiver entre **= “ ”** pelo nome desejado. Se eu quisesse renomear o primeiro preset como Flamengo, o texto ficaria **PresetName=“Flamengo”**

O mesmo se aplica se você quiser mudar o valor de um parâmetro, basta achar o parâmetro e mudar o número entre os sinais **=“ ”**