

Trabalho de Composição Algoritma

Resumo:

É um programa em C++ com auxílio da biblioteca MuM que sorteia notas aleatórias com pesos que se baseiam em probabilidades já estabelecidas relacionadas à nota anterior e a uma sequência de acordes dada pelo usuário, e com essas notas cria uma música.

O objetivo é observar se música aleatória com estruturas tonais atreladas aos pesos usados nessa aleatoriedade pode fazer uma música ser aceitável como música tonal.

As Entradas:

- Uma sequência de acordes, no formato de cifra, separados por espaço. Os acordes aceitos são C, C7, Cm, Dm, D7, Em, E7, F, G, Am, A7 e Bb.
- Um inteiro que indica quantas vezes repetir, 0 vezes da erro.
- Um inteiro que indica quantos tempos por compasso, deve ser um valor maior que 1, além disso recomenda-se o valor 3 (ritmo ternário) ou valor 4 (quaternário), mas valores como 5 ou 7 também são válidos como teste.
- Um inteiro que indica a quantidade de batimentos por minuto.

A Harmonia:

Com essa sequência de acordes é criada uma harmonia na voz 1 e com o instrumento 1 da orquestra preparada (que é o mesmo da orquestra padrão da biblioteca MuM), o compasso é tempos/4, onde o número de tempos do compasso é dado pelo usuário, no primeiro tempo do compasso é colocado o baixo com amplitude maior e nos outros tempos são colocadas as notas do acorde juntas e com amplitude menor. As notas da harmonia variam de 43 até 60 na notação de nota midi.

É importante resaltar que essa harmonia segue o tom de dó maior ou lá menor, então é possível pensar em graus, desde que o acorde inicial seja C ou Am. Por exemplo, se quisermos uma sequência de acorde assim: F Am Bb C, deveremos transpor eles para dó maior, ou seja, I III IV V, que seria C Em F G.

A Melodia:

A partir dessa harmonia e uma nota inicial, começa a criação da melodia na voz 0 com o instrumento 4 da orquestra criada, esse instrumento foi criado por síntese aditiva.
Escolha da nota:

Existe uma matriz de pesos, que diz qual a chance de certa nota acontecer dependendo do acorde atual e da nota anterior, a soma dessas chances entre todas as notas é 100. Por exemplo o pensamento é assim: qual a chance do fá mais próximo acontecer dado que estamos em Bb e a nota

anterior é um sol, nesse caso foi 30%.

Com essa matriz dando os pesos para as notas ocorrerem, pode começar uma escolha aleatória da próxima nota segundo esses pesos. Isso é feito com uma urna, que é preenchida a cada nota com os índices das notas(0 a 11)segundo os pesos das notas a serem escolhidas, essa urna é embaralhada, e é escolhido um numero aleatório dela. Lógico, quantos mais de uma certa nota estiver na urna, maior é a chance de ela ser escolhida.

Controle da oitava:

Ainda há um controle da oitava em que a nota será tocada. A informação da nota escolhida so tem um número de 0 a 11 indicando a nota em uma só oitava, então há uma variável que controla a oitava de tal forma que não fique grave ou agudo demais, ainda varia para deixar a melodia com menos saltos, por exemplo se estamos em dó(0) e a próxima nota escolhida for si(11), essa variável abaixa uma oitava, ou seja se for na oitava 72, o dó será o número $0+72$ e o si será o $60+11$, que é 71.

Escolha da duração:

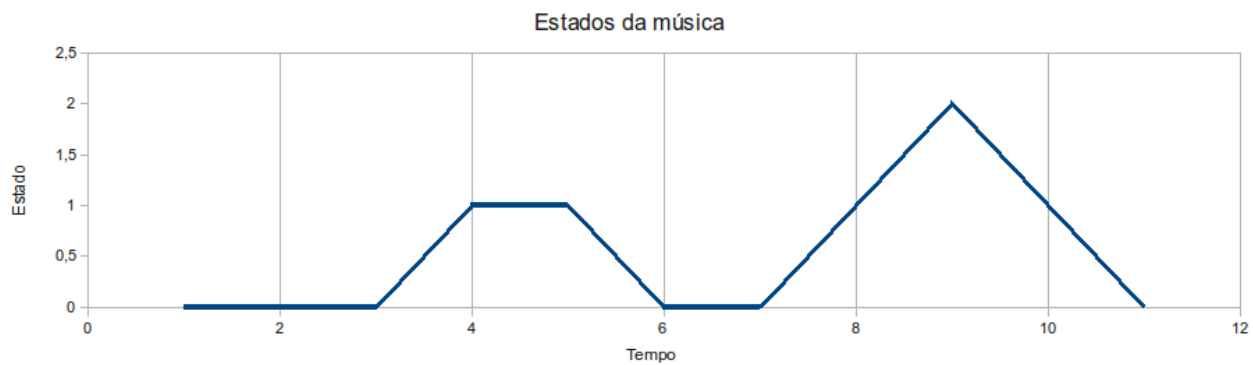
Como a forma como as notas são escolhidas já é predefinida, a única coisa que se pode mudar livremente é a duração delas, de fato ainda fiz algo parecido, algo aleatório mas com uma estrutura por trás.

Foram criadas 5 funções que determinam a duração das notas, algumas delas são mais aleatórias que outras, mas todas recebem 2 parametros: se a nota pertence ao acorde e se a música está no estado rápido. Se a nota pertence ao acorde as funções de duração tendem a deixá-las mais demoradas; se estamos no estado rápido, as funções tendem a deixar a duração da nota mais demorada. As funções sao escolhidas aleatoriamente, mas também com pesos diferentes para serem escolhidas. Ainda há um temporizador que faz com que sequencias de notas sigam a mesma função, então as notas não ficam soltas seguindo uma função de cada vez, mas sim cada trecho(inicialmente de 3 a 8 e variando de 9 a 14) siga a mesma função, dando um sentido maior a estruturação das duração ads notas.

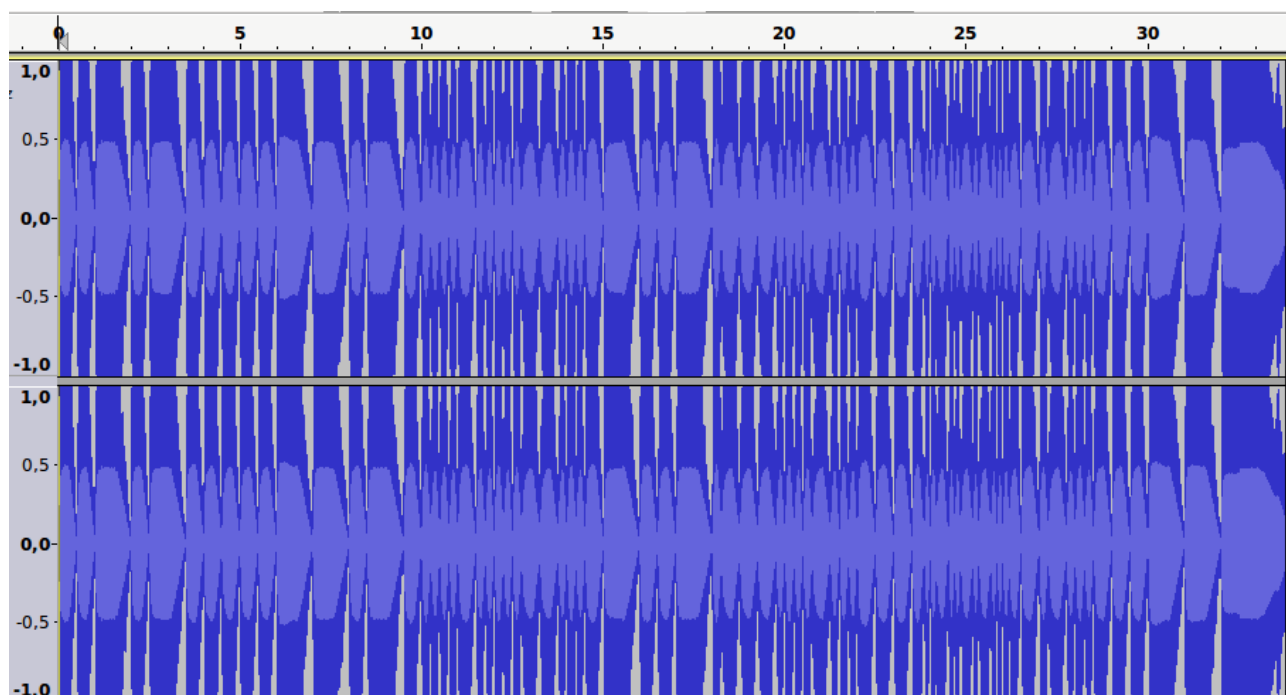
O estado rápido:

A música foi estruturada para que não fique só na mesma o tempo todo, foi feita para que se perceba um inicio, algum crescimento e algum final, como em boa parte das músicas. Então foi criado esse estado rápido, que faz com que as notas tenham uma tendencia a serem mais rapidas, além de aumentar os limites de alturas totais das notas, o que pode fazer com que a música fique mais aguda por alguns trechos, nesse estado ainda aumenta o número de notas que seguem a mesma função de duração, pois ja que as notas estão mais rapidas, mais notas devem ter a mesma logica de escolha de duração.

Então a musica foi dividida em 11 partes iguais, de tal forma que quem em algumas dessas partes estarão no estado normal e outras no estado rápido. Foi escolhido arbitrariamente que a música estará no estado normal na primeira, segunda, terceira, sexta e decima primeira partes dessas 11 partes, nos outros momentos estará no estado rápido. Há ainda um estado 'ultra rápido' que acontece na nona parte, nesse estado se acentua ainda mais as variações que o estado rápido causa.



Vizualização das mudanças de estado.



O final:

Por último é acrescentado mais um compasso ao final. Na harmonia é colocada o primeiro acorde com o acorde. Na melodia é escolhida uma nota(que dura o compasso todo) que pertence ao acorde de tal forma que a distancia entre ela e a última nota escolhida aleatoriamente seja mínima, assim há a sensação de que ela resolveu naquele acorde, ou seja, uma sensação de finalização.

Assim a melodia está pronta. Então só ocorre uma compressão do material inteiro para seguir o número de batimentos por minuto estabelecidos pelo usuário.

Por fim ela é exportada como melodia.sco, o arquivo de partitura do csound, e aberta no csound com junto com a orquestra orq.orc, ja fornecida.

Esquema de como funciona o programa.

