

Lista de requisitos do Keytone

Keytone é um software que tem como objetivo a conversão de um texto qualquer (a princípio livre/não estruturado) em música, a partir dos diversos tipos de comandos presentes neste texto.

1. No geral, o software deverá receber como entrada uma sequência de caracteres *string* e, baseado numa lista pré-definida, emitir comandos para executar os padrões sonoros desejados.
2. O programa deverá ser capaz de diferenciar os variados comandos emitidos por cada tecla, possuindo uma seção que lê cada uma e realiza as seguintes operações presentes:
 - 2.1. As letras maiúsculas de A até H são dedicadas a emitir certas notas com o instrumento atual, devendo ativar a nota selecionada.
 - 2.2. As letras minúsculas de a até h são teclas de silêncio ou pausa: isso significa que elas não devem emitir som.
 - 2.3. A tecla espaço deverá aumentar o volume para o seu dobro.
 - 2.4. Os sinais !, u, U, O, o, consoantes restantes, sinal nova linha, ;, dígitos de valor ímpar e o caractere vírgula devem trocar para os respectivos instrumentos listados no PDF relacionado.
 - 2.5. Os dígitos pares devem executar a troca do instrumento atual para aquele que está a frente pelo número de dígitos, não trocando em caso de zero.
 - 2.6. ? e . deverão alterar a frequência do som em uma oitava, retornando ao estado default se já estiver no limite.
 - 2.7. Para qualquer outro caso, o sistema deverá:
 - 2.7.1. Repetir a nota que foi tocada imediatamente antes ou:
 - 2.7.2. Fazer uma pausa.
 - 2.7.3. Ademais, é preciso sempre carregar o registro de qual foi a tecla anterior e manter este registro atualizado.
 - 2.8. Quaisquer outros comandos a serem adicionados precisam ser devidamente listados de forma que não haja confusão com outros comandos já acionados.
3. O sistema deverá ser capaz de produzir algum tipo de registro sobre o que cada comando executa em modo de notações de partitura, melhorando a clareza dos padrões musicais presentes.
4. O software deverá ser capaz de gerar texto a partir de inputs inseridos diretamente em sua interface (textos por arquivo são opcionais).
5. O sistema deverá ser capaz de emitir os sons do software de uma maneira fluida e sem que haja interferência indevida entre os diferentes elementos (ou seja, sem que comandos sejam ignorados).
6. Idealmente, o programa deverá funcionar de modo que não interfira de maneira negativa com outros elementos da máquina em que está rodando.

7. O operador deve também ser capaz de modificar as condições iniciais do sistema antes de tocar o seu texto (por exemplo, definir volume, oitava, até ponto de início do som se assim desejar).
8. O volume e a oitava do som gerado pelo texto devem ser capazes de ser alterados via recursos além dos comandos.
9. Haverá botões de reset, pause, e stop na parte inferior da interface, também sendo possível ativá-los independente do conteúdo do texto.
10. O programa deverá ter uma interface simples, tendo a postos uma aba que mostre os diferentes comandos.
11. No geral, o sistema deverá ser o mais simples para o uso de seus usuários o possível.