

Utilização do Ambiente Mosaicode como ferramenta de apoio para o ensino de Computação Musical

Flávio Luiz Schiavoni, Thiago Thadeu Souto Cardoso,
André Lucas Nascimento Gomes, Frederico Ribeiro Resende, José Mauro da Silva Sandy

¹ Departamento de Computação (DCOMP)
Universidade Federal de São João del-Rei
São João del-Rei – Minas Gerais – Brazil

fls@ufsj.edu.br, souto.t@hotmail.com

andgomes95@gmail.com, fredribeiro97@gmail.com, jmsandy@gmail.com

Abstract. *The Computer Music is an interdisciplinary area that involves knowledge of several other areas. Due to the practical work, the teaching of Music Computing may depend on previous knowledge in algorithms and programming. To help novices to implement computer music applications, this work presents the use and evaluation of the Mosaicode, a visual programming environment, as an educational support tool for teaching Computer Music.*

Resumo. *A Computação Musical é uma área interdisciplinar que envolve conhecimento de diversas outras áreas. Devido aos trabalhos práticos, o ensino da Computação Musical pode depender de conhecimentos prévios em algoritmos e programação. Para auxiliar usuários sem conhecimento em programação a implementar aplicações musicais, este trabalho apresenta a utilização e avaliação do Mosaicode, um ambiente de programação visual, como ferramenta de apoio educacional para o ensino de Computação Musical.*

1. Introdução

A **Computação Musical** é uma área interdisciplinar da computação que trabalha a fronteira entre computação e música. Analisando este domínio um pouco mais a fundo, perceberemos que o mesmo é considerado interdisciplinar, por envolver também o conhecimento de diversas outras áreas como Engenharia Elétrica, Engenharia de Áudio, Física, Matemática, Música, Psicologia e Artes em geral [Miletto et al. 2004a]. Além disso, dentro da própria computação, esta área envolve outras áreas como as Realidades Virtual e Aumentada, Processamentos de Sinais, Interface Humano Computador, Sistemas distribuídos e de alto desempenho, dentre outras [Schiavoni and Costalonga 2015].

O ensino de Computação Musical costuma envolver o conhecimento técnico para problemas computacionais cujas soluções utiliza algoritmos de síntese de sons, aritmética de sinais, filtros de áudio, efeitos musicais digitais, criação de instrumentos musicais, processamento e armazenamento de som, desenvolvimento de aplicações para compositores e artistas digitais [Miletto et al. 2004a].

Como lembra Miller Puckette, autor do Pure Data: “O processo de fazer música computacional é, primeiro, escrever um software, e então fazer música com ele” - [Puckette 2002]¹. Por esta razão, o ensino de Computação Musical passa por uma di-

¹“The process of doing computer music is, first, to write software, and then to make music with it.”

ficuldade inerente ao ensino de computação que é o ensino de programação sendo que o objetivo de um curso de Computação Musical não é ensinar a programar, mas:

Introduzir ao aluno o ferramental teórico e prático para o uso de computadores em produções musicais. Este ferramental deve permitir ao aluno analisar, processar e sintetizar sons e estruturas musicais com o apoio de computadores. Incluindo o conhecimento básico de processamentos de sinais e áudio digital, algoritmos clássicos de filtro e síntese, sinais de controle (MIDI e OSC), conceitos de processamento em tempo real e composição apoiada por computadores.²

Assim, torna-se uma política didática educacional adotar uma estratégia que simplifique a realização dos trabalhos práticos de computação musical e que permita o desenvolvimento das técnicas de síntese, controle e processamento por alunos iniciantes em computação como artistas e músicos. Para contornar esta dificuldade, diversas iniciativas surgiram para simplificar a programação por leigos como linguagens de programação musical e ambientes de programação visual para o domínio da computação musical. A Seção 2 deste documento traz uma visão geral destas linguagens e ambientes de programação.

Visando trazer mais uma opção para o apoio educacional e para o ensino de programação para leigos, este trabalho apresenta o ambiente de programação visual Mosai-code, desenvolvido pelos autores deste trabalho. Este ambiente, apresentado na Seção 3, permite o desenvolvimento de aplicações musicais e permite que leigos criem aplicações sem necessitar de grandes conhecimentos em computação.

Este ambiente foi utilizado como ambiente de apoio para a disciplina de Introdução à Computação Musical, uma disciplina optativa do Curso Bacharelado em Ciência da Computação da Universidade Federal de São João del-Rei, ofertada no primeiro semestre de 2018. Durante este semestre, alunos de computação e da música, com conhecimentos variados em programação e música realizaram tarefas utilizando este ambiente. O histórico desta disciplina está descrito na Seção 4 deste documento. A avaliação deste ambiente por estes alunos é apresentada na Seção 5 e o resultado desta pesquisa é apresentado na Seção 6.

Este trabalho traz ainda na Seção 7 uma discussão entre a eficácia dos resultados, comparando a utilização da ferramenta exposta na Seção 3 com outras linguagens utilizadas durante a disciplina.

2. Trabalhos Relacionados

Aliar o ensino de computação musical e sua teoria com exemplos práticos de programação é uma estratégia adotada em alguns materiais didáticos da área.

O livro “Elements of Computer Music” de Franklin Richard Moore [Moore 1990] possui diversos exemplos desenvolvidos na linguagem CMusic³, de mesmo autor. A linguagem CMusic, originalmente lançada em 1990, foi juntamente com as linguagens CSound e CMix [Pope 1993] desenvolvida para facilitar a codificação de sons e a síntese

²Retirado do Plano de Ensino da disciplina Introdução à Computação Musical do Curso de Ciência da Computação da Universidade Federal de São João del-Rei, disponível em: <https://ufsj.edu.br/ccomp/documentos.php>.

³Disponível em <http://crca-archive.ucsd.edu/~carl/>.

musical. Estas linguagens possuem foco na implementação de algoritmos de processamentos de sinal e são voltadas para a criação musical [Garton and Topper 1997].

O livro “The Audio programming Book” [Boulanger and Lazzarini 2010] dos autores Richard Boulanger e Victor Lazzarini utiliza a linguagem CSound⁴, desenvolvida por Barry Vercoe [Vercoe 1986] para ilustrar seus exemplos.

O livro “Theory and Techniques of Electronic Music” do Miller Puckette possui exemplos usando o ambiente de programação visual Pure Data, ou simplesmente Pd⁵ [Puckette 2007]. O Pd é ambiente de programação visual em tempo real desenvolvido pelo mesmo autor que utiliza o conceito de trabalhar com “dados puros”, permitindo a integração de diversos tipos de dados como áudio, vídeo, imagem e MIDI. Cada objeto é uma unidade que executa uma funcionalidade mínima, que é processar e transmitir dados musicais, matemáticos, gráficos, dentre outros. Mesmo passado mais de duas décadas de sua criação, ele ainda possui muitos desenvolvedores de *patches* ativos, propiciando uma forte comunidade de suporte.

Já o livro “DAFX: Digital Audio Effects” [Zölzer 2011] de Udo Zölzer, disponibiliza seus exemplos de processamentos de sinais para a linguagem MatLab⁶. Estes exemplos podem ser facilmente adaptados para a linguagem de computação Octave⁷, similar ao MatLab mas de código livre.

Além destas linguagens, que são amplamente utilizadas para apoiar o ensino de computação musical por serem disponibilizadas com material didático, há a possibilidade de utilizar outras linguagens de programação musical, como SuperCollider⁸ ou FAUST⁹ ou ainda linguagens de propósito geral, como Python ou C.

3. O ambiente de programação Musical Mosaiccode

O Mosaiccode é um ambiente de programação visual para criação de aplicações de linguagens de domínio específico. Este ambiente permite criar por meio de diagramas protótipos funcionais de aplicações voltadas para algum destes domínios. Este ambiente é um gerador de código e é possível a visualização do código gerado a qualquer etapa desta construção. Cada domínio de aplicação atendido por este ambiente de programação é disponibilizado na ferramenta como uma extensão sendo que as extensões são construídas através de uma biblioteca ou API. Cada extensão é composta por um conjunto de Blocos e suas propriedades, Conexões e um Padrão de geração de código.

Um Bloco é a uma unidade de código que simboliza uma funcionalidade mínima do domínio em questão. Um bloco pode possuir um conjunto de propriedades estáticas definidas em tempo de programação que possibilitam a configuração do código gerado, podendo possuir também propriedades dinâmicas que são configuradas por outros Blocos chamadas de Portas.

A ligação entre a porta de saídas e entradas de Blocos distintos são chamadas Conexões. As Conexões definem o fluxo de comunicação de uma determinada combinação

⁴Disponível em <https://csound.com/>.

⁵Disponível em <http://puredata.info/>.

⁶Disponível em <https://www.mathworks.com/products/matlab.html>.

⁷Disponível em <https://www.gnu.org/software/octave/>.

⁸Disponível em <https://supercollider.github.io/>.

⁹Disponível em <http://faust.grame.fr/>.

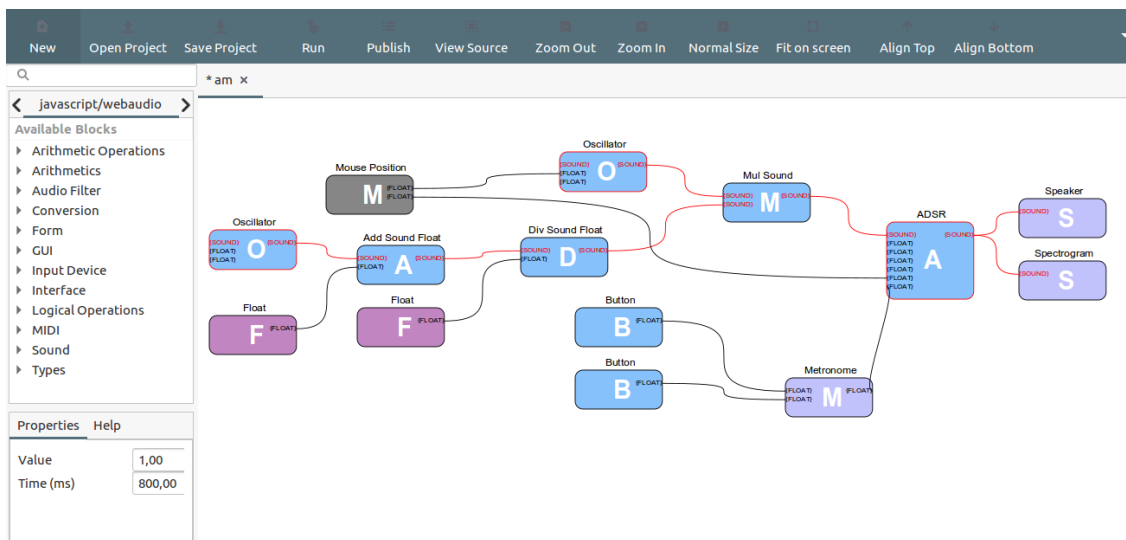


Figura 1. Ambiente Mosaicode e um diagrama criado dentro da extensão Javascript / Webaudio.

de Blocos definindo a ordem em que cada um deve ser gerado no código fonte. Esta combinação é chamada de Diagrama e é através de um Diagrama que é gerado o código de uma aplicação criada no ambiente. Desta forma, um Bloco pode receber informações, realizar seu devido processamento e por fim, repassar o processamento realizado e os dados gerados para o bloco seguinte no qual está conectado.

Além de gerar o código a partir dos seus Blocos e Conexões, o Mosaicode permite também a execução do código criado, para facilitar ainda mais e permitir que o usuário não precise compilar e executar em separado o diagrama criado a cada vez que desejar. Na Figura 1, podemos visualizar o ambiente Mosaicode e um diagrama criado no mesmo.

O Mosaicode conta atualmente com Extensões para a geração de códigos em C/C++ para Síntese de imagens, utilizando a API OpenGL, visão computacional, utilizando a API OpenCV, síntese e processamento de sons, por meio de uma API própria e também por uma extensão em Javascript baseada na API webaudio.

3.1. Extensão JavaScript - WebAudio

Dentre as extensões criadas para o Mosaicode, iremos abordar a que trata de síntese musical, através da API Javascript/Webaudio. Voltada para o desenvolvimento de WebArt, nesta extensão, o fluxo de dados transmitido através dos Blocos é o *SOUND*, que pode ser originado de arquivos de áudio, microfones ou de Blocos de síntese, como ruídos e osciladores.

A extensão possui várias categorias de Blocos, como Fontes de Som, Filtros de Áudio, Operações Lógicas e Aritméticas e Dispositivos de Entrada. Na Figura 2, temos um exemplo simples de um diagrama construído utilizando a extensão Javascript/Webaudio.

Há ainda Blocos baseados na API Webmidi, que trazem consigo a possibilidade de detectar entradas de dispositivos que utilizam deste protocolo, permitindo a manipulação e controle de notas, a partir de teclados e outros objetos MIDI. Esta extensão, que

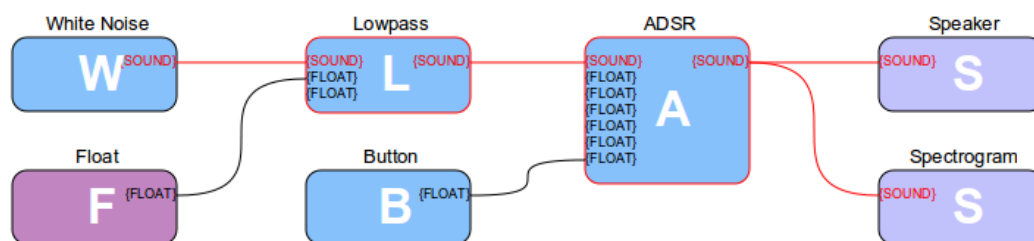


Figura 2. Diagrama utilizando Blocos de ruído, filtragem e envelopamento dentro da extensão Javascript/Webaudio.

conta ainda com blocos para visualização do sinal, GUI para entrada de dados, entrada de dispositivos físicos como mouse, teclado e sensores de celulares entre outras, é o objetivo desta pesquisa e o foco deste artigo. Apesar de haver críticas quanto a utilização da API webaudio para o desenvolvimento de aplicações musicais [Lazzarini et al. 2015], a mesma pode ser utilizada para o desenvolvimento de aplicações ubíquas na World Wide Web [Lazzarini et al. 2014].

4. O Curso de Computação Musical na UFSJ

A disciplina de Introdução à Computação Musical foi criada no Curso de Ciência da Computação da Universidade Federal de São João del-Rei em 2014. Esta disciplina possui como conteúdo a história da computação musical, como realizar síntese de áudio, aritmética de sinais, visualização e análise de sinais, criação de envelopes e efeitos musicais, utilização do protocolo MIDI para construção de música simbólica. No final da matéria, com o conhecimento adquirido durante o semestre, o aluno tem como tarefa construir um instrumento musical digital utilizando diferentes técnicas de síntese e aplicando efeitos de áudio.

Desde sua criação, há interesse de oferecer esta disciplina para alunos de outros cursos interessados em criações musicais, como o Curso de Música, Artes Cênicas, Arquitetura, Engenharia Elétrica, Artes Aplicadas e outros. No entanto, a necessidade de possuir um conhecimento prévio em Computação e programação sempre tornou proibitivo este oferecimento interdisciplinar.

No primeiro oferecimento desta disciplina, em 2014, optou-se pela utilização do ambiente Pure Data como ambiente de apoio ao desenvolvimento dos trabalhos práticos. Este ambiente se mostrou bastante interessante como ferramenta de apoio da disciplina, no entanto, devido ao público da disciplina ter sido alunos do Curso de Ciência da Computação, diversos alunos demonstraram bastante dificuldade de entender a programação visual deste ambiente. Também houve uma grande dificuldade para os alunos em abstrair a programação em Pure Data e entender como seria possível integrar os conceitos aprendidos neste curso com outras linguagens de programação ou outras aplicações. Assim, este ambiente mostrou-se de difícil entendimento para os próprios alunos de computação acostumados com programação textual.

Tentando solucionar esta questão, na segunda vez em que esta disciplina foi ofertada, em 2015, foi utilizado a linguagem C e APIs de desenvolvimento como libsoundfile e portaudio. Esta tentativa foi um tanto frustrante pois os alunos empregaram muita energia resolvendo problemas da linguagem C em vez de se empenhar em entender as técnicas

de computação musical. Além disto, integrar interfaces de controle de usuários (GUI) ou interfaces externas requeriam o aprendizado de várias outras bibliotecas e APIs o que tornou tal integração praticamente impossível.

Em 2016 esta disciplina foi novamente oferecida e desta vez foi utilizado como apoio a API Web Audio em JavaScript. O uso de javascript para a programação musical permitiu o desenvolvimento de aplicações sem o requerimento de nenhuma ferramenta em especial, já que tais códigos podem ser executados em qualquer navegador de Internet, além de permitir integrar de maneira simples componentes gráficos para a interação do usuário. Ainda assim, tal trabalhos práticos exigiam o aprendizado de uma linguagem de programação.

A tentativa de lecionar este curso sem a exigência de programação levou a construção deste ambiente de programação visual, iniciada em 2014 e ainda não concluída. Esta primeira oportunidade, em 2018, de ensinar o conteúdo de Computação Musical e desenvolver as práticas requeridas no ambiente Mosaicode foi um grande desafio para a equipe envolvida com o desenvolvimento deste ambiente.

5. Metodologia

Segundo [Miletto et al. 2004b], os programas de computador devem ser visto para auxiliar a prática do ensino e enriquecer o ambiente de aprendizado e não para substituir o professor. E é neste contexto que iremos avaliar o ambiente de programação Visual Mosaicode como ferramenta de apoio na disciplina de Introdução a Computação Musical do Curso de Ciência da Computação da Universidade Federal de São João del-Rei para a execução de uma parte dos trabalhos práticos deste curso no primeiro semestre de 2018. Esta disciplina foi assistida por alunos de dois cursos diferentes: alunos do curso de computação e alunos do curso de música totalizando quase 20 alunos regularmente inscritos. Para este semestre não foi exigido o cumprimento de outra disciplina, como Algoritmos e Estruturas de Dados, para a matrícula no curso de Introdução à Computação Musical.

As aulas foram iniciadas de maneira teórica e expositivas e depois de alcançado um nível inicial de conhecimento sobre processamento de sinais foram ministradas aulas práticas para validar e experimentar o conhecimento teórico adquirido. Nas aulas práticas foram propostos vários exercícios e implementações para os alunos fazerem utilizando o ambiente de programação Mosaicode e no meio do curso foi aplicado um questionário sobre alguns quesitos de avaliação.

Esta avaliação utilizou um formulário online com 26 perguntas direcionada aos alunos que cursaram esta disciplina. As duas primeiras perguntas tinham por objetivo verificar se nosso entrevistado era realmente o nosso aluno em questão.

1. Você usou o Mosaicode nas aulas de Computação musical?
2. Em que frequência utiliza o computador diariamente?

As próximas 6 perguntas serviram para verificarmos o perfil dos entrevistados e entender qual o nível de conhecimento destes para atender os exercícios práticos requisitados nesta disciplina.

- Qual seu nível de conhecimento em
 3. Computação?

4. Programação?
5. Música?
6. Criação musical?
7. Computação Musical antes das aulas utilizando Mosaicode?
8. Computação Musical depois das aulas utilizando Mosaicode?

Depois disso, seguiu-se 2 perguntas para entender o nível de dificuldade relacionado a este ambiente ser utilizado como ferramenta complementar deste curso.

- Nível de dificuldade
 9. Para utilizar o Mosaicode?
 10. Para fazer os exercícios com Mosaicode propostos em aula?

As próximas perguntas focaram no esforço para realizar tarefas específicas deste curso:

- Esforço em Tarefas Específicas
 11. Fazer um som qualquer?
 12. Fazer uma síntese aditiva?
 13. Fazer uma síntese subtrativa?
 14. Fazer uma síntese FM?
 15. Fazer uma síntese AM?

O próximo grupo de perguntas buscava entender onde exatamente estaria a dificuldade encontrada em aplicar os conceitos de computação musical, como o de síntese, em um trabalho prático no ambiente.

- O Mosaicode
 16. É intuitivo como a ferramenta funciona?
 17. Possui interface é amigável?
 18. Poderia ser utilizado no próximo semestre?
 19. Atendeu os propósitos estabelecidos?
 20. Apresentou um código gerado que propiciasse aprendizado?
 21. Ajudou a entender os conceitos teóricos de Computação Musical?

Aproveitou-se nesta avaliação para verificar também os critérios de usabilidade da ferramenta e a relação entre dificuldade teórica e prática para a execução das tarefas requisitadas.

- Usabilidade do Mosaicode
 22. Conseguiria utilizar a ferramenta sem o auxílio do professor?
 23. Conseguiria utilizar a ferramenta sem conhecimento prévio de programação?
 24. Conseguiria utilizar a ferramenta sem conhecimento teórico prévio em música?
 25. Conseguiria utilizar a ferramenta sem conhecimento prévio em Computação Musical?
 26. O acesso ao código gerado auxiliou o aprendizado?

Tabela 1. Identificação do público alvo.

	Pergunta 1	Pergunta 2
Nada / Muito pouco	0%	0%
Quase nada / pouco	0%	12,5%
Ajudou / Moderado	37,5%	0%
Quase tudo / Quase sempre	50%	31,3%
Tudo / Sempre	12,5%	56,6%

6. Resultados

De acordo com os dados apresentados na Tabela 1, os alunos participantes do questionário possuem acesso à computadores. A resposta destas questões mostram também que a ferramenta serviu de auxílio para a maior parte da disciplina. Vale notar que alguns trabalhos práticos deveriam ser feitos utilizando outras ferramentas como o editor de áudio Audacity ou o sequenciador LMMS.

O próximo grupo de perguntas, cujas respostas estão apresentadas na Tabela 2, mostra que os alunos participantes da enquete consideram que possuem um conhecimento moderado a satisfatório em computação e programação em geral. Quanto em relação a compreensão sobre música e composição musical, a maioria considera seu entendimento fraco ou moderado, embora tenha um número considerável de pessoas que consideram ter um conhecimento musical.

Tabela 2. Conhecimento dos alunos sobre assuntos acerca da disciplina. As colunas 3 a 8 representam as respostas para estas perguntas no questionário aplicado.

	3	4	5	6	7	8
Fraco	0%	6,25%	37,5%	50%	68,75%	6,25%
Moderado	18,75%	31,25%	18,75%	25%	25%	50%
Satisfatório	62,5%	56,25%	31,25%	25%	6,25%	31,25%
Muito Bom	18,75%	6,25%	12,5%	0%	0%	12,5%
Excelente	0%	0%	0%	0%	0%	0%

Podemos observar nas colunas **7** e **8** da Tabela 2 a evolução do conhecimento dos alunos sobre computação musical depois do uso da ferramenta. É possível perceber que os alunos alcançaram avanços significativos quanto ao nível de aprendizagem do domínio em questão utilizando a ferramenta. A grande maioria dos alunos iniciou seus estudos com pouco ou nenhum conhecimento em Computação Musical e passou a ter um estado moderado ou satisfatório neste mesmo quesito após o uso da ferramenta, o que significa bons avanços considerando que esta foi a primeira tentativa didática do Mosaicode.

A Tabela 3 apresenta as questões relacionadas a utilização da ferramenta. Nela, é possível observar como o Mosaicode possui um potencial para se tornar uma ferramenta bastante prática e intuitiva para novos usuários. Mais de 60% dos alunos definiram o ambiente com uma usabilidade simples e totalmente funcional para as atividades requeridas, seja em sala de aula, ou nos trabalhos práticos.

Na Tabela 4 foi medido o esforço do usuário ao utilizar a ferramenta quando o

Tabela 3. Nível de dificuldade na utilização da ferramenta. As colunas 9 e 10 representam as respostas para estas perguntas no questionário aplicado.

	9	10
Fraco	12,5%	25%
Moderado	56,25%	37,5%
Satisfatório	18,75%	12,5%
Alto	12,5%	18,75%
Muito Alto	0%	6,25%

professor solicitou alguma implementação específica. Estas perguntas tiveram como respostas que a ferramenta exigiu pouco esforço ou esforço moderado do usuário para a realização das tarefas específicas propostas da disciplinas.

Tabela 4. Esforço em tarefas específicas. As colunas 11 a 15 representam as respostas para estas perguntas no questionário aplicado.

	11	12	13	14	15
Nenhum esforço	50%	0%	50%	68,75%	0%
Pouco esforço	37,5%	37,25%	12,5%	25%	12,5%
Esforço moderado	12,5%	56,25%	56,25%	6,25%	62,5%
Esforço grande	0%	6,25%	18,75%	0%	18,75%
Esforço muito grande	0%	0%	12,5%	0%	6,25%

Em concordância com os dados obtidos pelo terceiro grupo de perguntas do questionário, a Tabela 5 reafirma em termos gerais a praticidade e eficiência da aplicação do Mosaicode nesta disciplina. Os dados mostram o quanto os alunos aceitaram a inovação da utilização de uma ferramenta para realizar os trabalhos práticos. A coluna **18**, que se refere quanto à utilização do Mosaicode na próxima vez em que a disciplina for oferecida, obteve 100% de apoio dos alunos, transparecendo que todos os alunos foram favoráveis à utilização e que o mesmo foi eficiente no que se foi exigido.

Tabela 5. Perguntas gerais sobre o ambiente Mosaicode. As colunas 16 a 21 representam as respostas para estas perguntas no questionário aplicado.

	16	17	18	19	20	21
Discordo Totalmente	0%	0%	0%	0%	0%	0%
Discordo	6,25%	6,25%	0%	6,25%	6,25%	0%
Não Sei	12,5%	6,25%	0%	6,25%	31,25%	50%
Concordo	56,25%	56,25%	37,5%	43,75%	37,5%	12,5%
Concordo Plenamente	25%	31,25%	62,5%	43,75%	25%	37,5%

É interessante notar que este é um ambiente de geração de código mas que, segundo as respostas da pergunta **20**, esta característica parece ter sido ignorada por grande parte dos usuários. Da mesma forma, é interessante verificar, segundo as respostas da pergunta **21**, que os usuários que conseguiram desenvolver as tarefas requisitadas com pouco ou nenhum esforço não tinham certeza sobre terem entendido os conceitos teóricos de Computação Musical.

A Tabela 6 apresenta algumas questões a serem trabalhadas futuramente. Grande parte dos alunos entrevistados afirmaram que não conseguiriam utilizar a ferramenta sem o apoio do professor. Isto pode revelar dois pontos: ou a ferramenta têm alguns pontos falhos quanto à sua intuitividade ou é impossível criar aplicações de Computação Musical sem um conhecimento mínimo do domínio de aplicação e sua teoria.

Ao que tange a programação, 80% dos alunos afirmaram que não precisariam de conhecimentos específicos de computação como programação para utilizar a ferramenta, algo que faz prova à capacidade do Mosaicode para abranger novos usuários que não estejam diretamente relacionados com computação (como artistas, performistas, músicos).

Tabela 6. Usabilidade do Mosaicode. As colunas 22 a 26 representam as respostas para estas perguntas no questionário aplicado.

	22	23	24	25	26
Discordo Totalmente	6,25%	0%	25%	25%	6,25%
Discordo	31,25%	0%	18,75%	25%	6,25%
Não Sei	31,25%	18,75%	18,75%	31,25%	43,75%
Concordo	12,5%	62,5%	12,5%	12,5%	37,5%
Concordo Plenamente	18,75%	18,75%	25%	6,25%	6,25%

7. Discussões

O resultado deste experimento em sala de aula possibilitou um fazer criativo que permitiu aliar o conhecimento técnico com a prática artística. Os alunos construíram aplicações musicais que poderiam levar um bom tempo de programação apenas inserindo e conectando Blocos e em poucos minutos. Isto permitiu uma maior experimentação do conteúdo dado, pois não há preocupação em construir códigos e consertar eventuais erros. Em diversos momentos o conhecimento técnico foi deixado de lado em favor de uma experimentação livre de códigos que podiam soar bem ou simplesmente não soar. Isto exigiu que o conteúdo programático fosse colocado de lado em favor de entender o que acontecia com o sinal quando alguns blocos do Mosaicode eram combinados de determinada maneira.

Os alunos também tiveram a oportunidade de recombinar os Blocos em diferentes posições, gerando diferentes códigos finais distintos. Se novamente compararmos o ambiente à programação, esta também é uma tarefa mais simples, pois reposicionar trechos de código pode ser uma atividade complicada, principalmente se tratando de aplicações complexas que exigem alocações de memória e cópia de dados em tempo real.

Os dados coletados nesta pesquisa indicam que o Mosaicode foi extremamente importante na velocidade do processo de aprendizagem de computação musical, pois, trabalhar inserindo e conectando Blocos é uma tarefa bem menos árdua do que programar textualmente (atividade que custaria mais tempo dedicado à lógica de programação do que para síntese musical essencialmente). Em diversos momentos os alunos extrapolaram os conceitos vistos em sala de aula e começaram a requisitar funcionalidades que o ambiente ainda não possuía além de um aprofundamento maior e um entendimento sobre questões de implementação que não estavam previstos no programa da disciplina.

Isto nos levou a reavaliar os blocos disponíveis em nosso ambiente de programação visual e, em muitas vezes, nos deparamos com dificuldades técnicas e requisitos não previstos inicialmente em nossa análise. Um exemplo disto foi a tentativa constante dos alunos em construir um sintetizador polifônico ou construir envelopes com comportamentos diferentes do ADSR. Em diversos momentos tivemos que reavaliar a nossa implementação e propor novos Blocos e soluções para resolver tais requisitos.

Além da vantagem aparente que o Mosaicode nos traz, como a facilidade oferecida pela programação visual, fazendo com que usuários que não tenham um conhecimento aprofundado do domínio consigam criar suas aplicações sem muitas dificuldades, este ambiente apresentou outras vantagens, como a possibilidade que os instrumentos construídos fossem compartilhados em rede. Isto permitiu aos alunos a cooperação e a experimentação de diversos instrumentos criados neste ambiente em sala de aula ou em outros contextos.

Uma questão interessante apresentada neste experimento é que houve anteriormente uma dificuldade em utilizar o Pure Data como ferramenta de apoio deste curso devido ao mesmo ser visual e neste experimento isto não aconteceu. Uma hipótese para esta mudança pode ser o fato de os alunos desta turma não terem obrigatoriamente passado pelas disciplinas mais avançadas de programação que antes eram pré-requisitos para Computação Musical e com isto estão mais abertos a programar visualmente. Esta questão exigirá uma pesquisa para verificar se esta hipótese é mesmo verdadeira.

8. Conclusão

A tentativa de ensinar Computação Musical sem a necessidade prévia de uma bagagem em programação parece um tanto utópica. A disciplina de Computação Musical requer trabalhos práticos de codificação para os alunos, trabalhos estes que lidavam por exemplo com sínteses de som, aplicação de efeitos e criação de instrumentos e isto pode ser um fator limitante e excludente para pessoas que não possuem um pré-requisito de programação.

Este artigo apresentou a utilização e a avaliação do ambiente de programação Visual Mosaicode como ferramenta de apoio para o curso de Computação Musical na Universidade Federal de São João del-Rei no primeiro semestre de 2018. Apesar de ser apenas uma primeira experiência, os resultados apresentados por esta pesquisa parecem bastante promissores e servem como motivação para a continuidade do desenvolvimento desta ferramenta.

Este ambiente conta ainda com extensões para desenvolvimento de aplicações na linguagem C usando APIs como openCV para visão computacional, OpenGL para síntese de imagem e uma API própria para processamento e síntese de som. Estas APIs não foram exploradas nesta pesquisa pois as mesmas não são multiplataforma e dependem da instalação de compiladores e bibliotecas para a execução do código gerado. Isto nos faz perceber que a utilização do webaudio é parte do sucesso deste experimento pois o código gerado costuma ser multiplataforma e fácil de ser distribuído.

Trabalhos Futuros

A utilização apresentada neste trabalho trouxe uma série de novas funcionalidades requisitadas pelos usuários do ambiente e que deverão ser incorporadas no futuro. Uma delas

é a capacidade de criar abstrações e novos Blocos utilizando a própria ferramenta, como ocorre no Pure Data, de maneira a permitir um reuso de código ainda maior dentro do próprio ambiente.

Os instrumentos criados pelos alunos neste curso deverão ser incorporados ao conjunto de aplicações da Orchidea - Orquestra de Ideias, um grupo de pesquisa e prática em artes digital que utiliza plataformas livres para criação artística.

Agradecimentos

Os autores gostariam de agradecer aos participantes do Grupo de Estudo em Arte Digital do Departamento de Computação da Universidade Federal de São João del-Rei e aos membros do Arts Lab in Interfaces, Computers, and Everything Else (ALICE.), que auxiliaram nos debates e desenvolvimento deste projeto. Também gostariam de agradecer todos os alunos da disciplina de Introdução a Computação Musical que ajudaram na pesquisa desse trabalho. Os Coautores gostariam de agradecer o apoio financeiro institucional da FAPEMIG, CNPq e da PROAE/PROPE/UFSJ.

Referências

- Boulanger, R. and Lazzarini, V. (2010). *The audio programming book*. the MIT Press.
- Garton, B. and Topper, D. (1997). Rtcmix-using cmix in real time. In *ICMC*.
- Lazzarini, V., Costello, E., Yi, S., et al. (2014). Development tools for ubiquitous music on the world wide web. In *Ubiquitous Music*, pages 111–128. Springer.
- Lazzarini, V., Yi, S., and Timoney, J. (2015). Web audio: Some critical considerations.
- Miletto, E. M., Costalonga, L. L., Flores, L. V., Fritsch, E. F., Pimenta, M. S., and Vicari, R. M. (2004a). Introdução à computação musical. In *IV Congresso Brasileiro de Computação*.
- Miletto, E. M., Costalonga, L. L., Flores, L. V., Fritsch, E. F., and Vicari, R. M. (2004b). Educação musical auxiliada por computador: algumas considerações e experiências. *RENOTE: revista novas tecnologias na educação [recurso eletrônico]*. Porto Alegre, RS.
- Moore, F. R. (1990). *Elements of Computer Music*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Pope, S. T. (1993). Machine tongues xv: Three packages for software sound synthesis. *Computer Music Journal*, 17(2):23–54.
- Puckette, M. (2002). Max at seventeen. *Computer Music Journal*, 26(4):31–43.
- Puckette, M. (2007). *The Theory and Technique of Electronic Music*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.
- Schiavoni, F. L. and Costalonga, L. (2015). Ubiquitous music: A computer science approach. *Journal of Cases on Information Technology (JCIT)*, 17(4):20–28.
- Vercoe, B. (1986). *Csound: A manual for the audio processing system and supporting programs with tutorials*. Massachusetts Institute of Technology.
- Zölzer, U. (2011). *DAFX: digital audio effects*. John Wiley & Sons.