Teste de Usabilidade do Sistema Mosaicode

Alternative Title: Usability Testing of Mosaicode System

Flávio Luiz Schiavoni
Universidade Federal de São João Del Rei – UFSJ
Departamento de Computação
São João Del Rei, MG, Brasil
fls@ufsj.edu.br

Luan Luiz Gonçalves
Universidade Federal de São João Del Rei – UFSJ
Departamento de Computação
São João Del Rei, MG, Brasil
luanlg.cco@gmail.com

RESUMO

O Sistema Mosaicode é um ambiente de programação visual que atendem o desenvolvimento de aplicações nos domínios de Arte Digital e Realidade Virtual. Por esta razão, esta ferramenta é caracterizada como uma linguagens de programação visual e de linguagens baseadas em domínio. Este trabalho apresenta um Teste de Usabilidade desse ambiente de programação com o intuito de avaliar a facilidade e eficiência de aprendizado e de uso, bem como satisfação do usuário e produtividade, características fundamentais para estas categorias de linguagens de programação. Os testes permitiram ainda a sugestão de futuras modificações e a identificação de problemas do sistema a serem solucionados.

Palavras-Chave

Mosaicode, VPL, DSL, Teste de Usabilidade, IHC

ABSTRACT

The Mosaicode System is a Visual Programming Environment that supports the development of applications on Digital Art and Virtual Reality domains. For this reason, this tool is characterized as a visual programming language and a specific domain language. This work presents a usability test of this programming environment in order to evaluate how easy and eficient it is to use and learn it, basic features for these categories of programming languages. The tests also pointed out suggestions of future modifications and an identification of system problems to be solved.

Keywords

Mosaicode, VPL, DSL, Usability Testing, HCI

Categories and Subject Descriptors

D.1.7 [**Software**]: PROGRAMMING TECHNIQUES—*Visual Programming*; D.2.4 [**Software**]: SOFTWARE ENGINEERING—*Software/Program Verification*

1. INTRODUÇÃO

As linguagens de programação visuais (VPL – *Visual Programming Language*) pertencem a uma categoria de linguagens que permitem ao usuário desenvolver sistemas usando uma notação bidimensional e interagir com o código por meio de uma representação gráfica em vez de editar um fluxo unidimensional de caracteres [4].

Essa categoria de linguagens baseia-se no princípio de que pode ser mais fácil entender gráficos do que textos e que, por isto, a programação por meio de diagramas pode trazer facilidade para o desenvolvimento de sistemas. Isto pode permitir que não programadores ou programadores iniciantes desenvolvam aplicações [3]. Além disso, a abstração de código por meio de diagrama pode trazer praticidade na alteração do código fazendo com que as mesmas sejam bastante adequadas para rápida prototipação.

Já as linguagens baseadas em domínio, DSL – *Domain-Specific (Programming) Language*, pertencem a uma categoria de linguagem de programação que permite que o usuário desenvolva sistemas dentro de um escopo bem definido. Diferente de linguagens de propósito geral, que podem ser utilizadas para o desenvolvimento de software para qualquer domínio de aplicação, essas linguagens facilitam o desenvolvimento de sistemas para um domínio específico mas muitas vezes restringem a utilização desta linguagem para o desenvolvimento de aplicações em outros domínios [7].

As DSL's atuam como uma biblioteca ou framework em uma linguagem de propósito geral e trazem implementadas várias funcionalidades de código específicas do domínio da mesma. Por essa razão, as vantagens potenciais das DSL's incluem custos de manutenção reduzidos por meio de reutilização de funcionalidades já criadas e maior portabilidade, confiabilidade, otimização e testabilidade [7]. Pela mesma razão, muitas vezes o usuário programador precisa ter conhecimento do domínio para utilizar a linguagem ou poderá não compreender a finalidade dos recursos da mesma.

Unindo a simplicidade de programação visual das VPL's com a reutilização de código das DSL's, apresentamos neste artigo a ferramenta Mosaicode, um ambiente de programação visual que pode ser utilizado para desenvolvimento de sistemas nos domínios específicos de arte digital e realidade virtual.

Com o intuito de validar se esse ambiente atende aos objetivos das duas categorias de linguagens de programação citadas anteriormente, VLP e DSL, este artigo apresenta sua avaliação de usabilidade. Com base nesta avaliação é apresentado propostas de melhorias com o intuito de aumentar

a usabilidade desse sistema.

Esta avaliação não será comparativa com outras VPL - DSL pois a) não foi encontrada uma linguagem para esses domínios que seja VPL, b) não foi encontrada uma quantidade razoável de usuários de uma linguagem DSL - VPL para música e c) o Mosaicode não é uma DSL mas permite que usuários criem DSLs.

Este artigo está organizado da seguinte forma: A Seção 2 apresenta conceitos relacionados com este trabalho, a Seção 3 apresenta a metodologia usada, a Seção 4 apresenta os resultados desta pesquisa e a Seção 5 apresenta a Conclusão.

2. TESTE DE USABILIDADE E IHC

Em linhas gerais, a área de Interação Humano-Computador investiga o "projeto (design), avaliação e implementação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos associados a este uso" [1, 5].

A avaliação de IHC avalia a qualidade do projeto de interface de uma ferramenta tanto ao longo do processo de desenvolvimento como quando o software está pronto [1], buscando com isso construir uma interface com alta qualidade.

Usabilidade é um conceito que descreve a qualidade da interação de usuários com uma interface, essa qualidade está associada aos princípios de [2] a) facilidade de aprendizado, b) facilidade de memorização de tarefas no caso de uso intermitente, c) produtividade de usuários na execução de tarefas, d) prevenção, visando a redução de erros por parte do usuário e e) satisfação subjetiva do usuário.

Entre as possíveis avaliações de IHC, o Teste de Usabilidade é um método de avaliação através de observação que permite ao avaliador coletar dados sobre situações que os participantes realizam suas atividades. A análise dos dados registrados permite identificar problemas que os participantes enfrentaram e problemas potenciais previstos pelo avaliador [1].

O método utilizado é composto por 5 atividades [1]: Preparação, Coleta de Dados, Interpretação, Consolidação dos Resultados e Relato dos Resultados.

3. METODOLOGIA

Está pesquisa utiliza o Teste de Usabilidade para apoiar o desenvolvimento do Mosaicode, apontando possíveis mudanças para melhorar a usabilidade. Com isso, busca-se identificar problemas reais que os participantes do Teste de Usabilidade enfrentaram, e não apenas problemas potenciais previstos pelo avaliador como em uma avaliação por inspeção.

Foi realizada uma preparação, definindo o perfil dos participantes, tarefas para os participantes, preparação do material e execução de um teste-piloto. Após a coleta de dados os mesmos foram interpretados, consolidados e relatados. O relato dos resultados apresenta problemas reais que os participantes enfrentaram.

Os testes foram aplicados separadamente para cada participantes, tendo que realizar duas tarefas e responder um questionário com intuito de registrar a opinião dos participantes no uso do sistema Mosaicode. Por meio da observação da experiência dos usuários pretendeu-se validar se o sistema em questão oferece poder expressivo para gerar sistemas, nas áreas de Computação Musical e Visão computacional. Além disso, busca-se validar o quanto esse ambiente pode tornar mais fácil e prático o desenvolvimento de sistemas específicos do seu domínio.

Após a execução das tarefas, os usuários responderam um questionário para a sugestão de mudanças na ferramenta. O questionário é um método familiar e acessível para avaliar a satisfação do usuário, sendo que, para a sua validade ser conferida se faz necessário realizá-lo em conjunto com Testes de Usabilidade ou com avaliação por especialistas [6]. Este item da avaliação permitiu aos usuários proporem melhorias para o sistema em relação a sua facilidade de uso e aprendizado.

Estas Atividades serão descritas individualmente a seguir.

3.1 Preparação

A atividade de **Preparação** no Teste de Usabilidade serve para definir as tarefas para os participantes executarem e o perfil dos participantes, além de preparar o material para observar e registar o uso e executa um teste-piloto.

Apesar do Mosaicode trabalhar de forma visual, arrastando e conectando blocos, é necessário um conhecimento prévio sobre Visão Computacional e/ou Computação Musical para sua utilização, uma vez que as funcionalidades dos blocos estão relacionados a essas áreas. Portanto, o perfil de usuário selecionado foi o de alunos dos curso de Ciência da Computação, que tenham sido alunos dessas disciplinas, para participar do teste. Para avaliar a experiência de aprendizado, é importante que os usuários não sejam experientes com o sistema, fazendo com que ocorra um processo de aprendizado durante a realização das tarefas, por isso, outra característica do perfil é de usuários leigos. Os participantes executaram as seguintes tarefas:

Tarefa 1

A primeira tarefa teve por objetivo avaliar o Mosaicode, utilizando a classe de plugins que geram códigos em *C/OpenCV*. Para isso, os participantes deveriam separar canais RGB da captura da câmera, aplicar atraso a estas imagens, recompôr os canais e apresentar o resultado.

Essa tarefa pode ser cumprida seguindo a seguinte sequência de passos: 1. Capturar a imagem da webcam em tempo real. Plugin: "Live Mode"; 2. Decompor o sinal de imagem. Plugin: "Decompose RGB"; 3. Atrasar o canal G em 10 frames e o canal B em 20 frames. Plugin: "Live Delay;" 4. Compor o sinal de imagem. Plugin: "Compose RGB"; 5. Mostrar imagem na tela. Plugin: "Show Image"; 6. Visualizar o código-fonte gerado e executa-lo.

A sequência de plugins utilizados e suas conexões é apresentada na Figura 1.

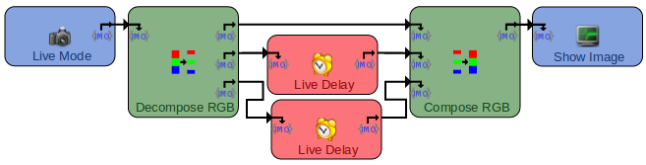

Figura 1: Imagem da tarefa 1

Tarefa 2

A segunda tarefa teve por objetivo avaliar o Mosaicode, utilizando a classe de plugins que geram códigos em *Javascript/Webaudio*. Para isso, os participantes deveriam mesclar o som de um oscilador com ruído branco, aplicar um envelope ADSR no sinal resultante e disparar o som deste

som com o clique de um botão, enviando tal sinal para a saída do sistema.

Esta tarefa pode ser cumprida seguindo os seguintes passos: 1. Adicionar o plugin "Oscillator"; 2. Adicionar o plugin "White Noise"; 3. Juntar os canais de áudio do "Oscillator"e do "White Noise". Plugin: "Channel Merger"; 4. Conectar a saída do Channel Merger em um envelope *ADSR*. Plugin: "ADSR"; 5. Adicionar um botão para acionar o envelope. Plugin: "Button"; 6. Conectar a saída do *ADSR* no plugin *Speaker*; 7. Executar.

Essa sequência de plugins e conexões pode ser visualizada na Figura 2.

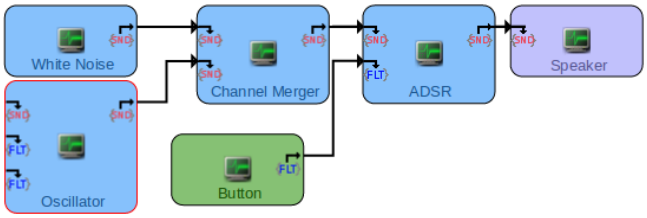

Figura 2: Imagem da tarefa 2

As duas tarefas são compostas pela mesma quantidade de plugins (6 blocos), variando apenas o número de conexões – a Tarefa 1 com 7 conexões e a Tarefa 2 com 5 conexões.

3.2 Coleta de Dados

A atividade de **Coleta de Dados** observa e registra a performance e a opinião dos participantes durante sessões de uso controlado.

Para a realização desta coleta, a ferramenta foi instalada em um laboratório de IHC da Instituição para a aplicação das tarefas aos participantes selecionados. Os dados coletados nos testes foram: 1. Tempo para conclusão da tarefa; 2. Número de usuários que não conseguiram realizar a tarefa; 3. Nível de dificuldade no uso do sistema; 4. Problemas encontrados durante os testes.

Esses dados foram coletados com um formulário a ser preenchido pelos participantes após o término da execução da avaliação com as perguntas apresentadas na Tabela 1.

Tabela 1: Formulário para os participantes.

-	Perguntas
1	Foi fácil aprender usar o sistema?
2	Penso que poderia me tornar produtivo rapidamente utilizando este sistema.
3	O sistema forneceu mensagens de erro que me disseram claramente como corrigir os problemas?
4	As informações fornecidas pelo sistema foram fáceis de compreender?
5	Gostei de usar a interface do sistema.
6	Este sistema tem todas as funções e capacidades que espero que tenha?

Foram dadas 7 opções de resposta com valores de 1 a 7 variando entre Discordo totalmente para o valor 1 até Concordo totalmente com o valor 7.

Esse formulário teve como objetivo avaliar a satisfação dos participantes levando em consideração a facilidade de uso e aprendizagem, produtividade, interface e recursos do sistema.

3.3 Interpretação e Consolidação dos resultados

As atividades de **Interpretação** e **Consolidação dos resultados** consistem em reunir, contabilizar e sumarizar os dados coletados dos participantes.

Devido a utilização de um formulário eletrônico criado unicamente para está avaliação, a tarefa de reunir os dados foi realizada de maneira automática pelo nosso formulário.

Com o intuito de contabilizar e sumarizar o tempo para a conclusão das tarefas foi calculado a média aritmética, conforme apresentado na Eq 1a, e o desvio padrão dos tempos coletados para cada tarefa, conforme apresentado na Eq 2a.

Fórmula da média aritmética:

$$\bar{X} = \frac{\sum x}{n}, \tag{1a}$$

onde: $\sum x$ é de todos os tempos coletados e n é número total de participantes.

Fórmula do desvio padrão:

$$\sigma = \sqrt{\frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n-1}}, \tag{2a}$$

onde: $\sum x^2$ é a soma do quadrado de cada um dos tempos coletados; $\sum x$ é a soma de todos os tempos coletados; e n é o número total de participantes.

Os dados coletados quanto aos "Problemas encontrados durante o teste" foram então analisados e sumarizados pelo avaliador sendo classificados pela sua gravidade [1] conforme apresentado a seguir:

- **Problema catastrófico:** impede que o usuário termine sua tarefa;
- **Problema sério:** atrapalha a execução da sua tarefa;
- **Problema cosmético:** atrasa a execução e/ou irrita usuários.

4. RESULTADOS

A última atividade de nossa avaliação trata-se do **Relato dos resultados** onde é relatado o desempenho a opinião dos participantes.

Tempo para conclusão da tarefa

A Tarefa 1 demorou, em média, 315 segundos para ser executada com um desvio padrão de 109 segundos, enquanto a conclusão da Tarefa 2 levou em média 245 segundos com um desvio padrão de 226 segundos, conforme apresentado na Tabela 2. Esta tabela traz ainda o resultado do teste realizado com um usuário experiente que demorou 63 segundos na Tarefa 1 e 40 segundos na Tarefa 2.

Tabela 2: Tempo de execução das Tarefas.

Tarefa	X	σ	Experiente
1	315	109	63
2	245	226	40

Observando a Tabela 2 é possível notar que a segunda Tarefa tomou menos tempo para sua execução do que a primeira. Isso pode dar um indício de que os participantes passaram a dominar melhor o ambiente, gastando menos tempo para a execução da tarefa. Nota-se que esta melhora com o tempo médio poderia ser ainda maior, como pode-se

observar pelo valor do desvio padrão. Na Tarefa 2, um participante demandou mais tempo para conclusão da mesma (em torno de 13 minutos), e essa demora aumentou consideravelmente a média do tempo para a execução dessa tarefa. Se o tempo desse usuário for excluído, o tempo de realização da segunda Tarefa é, em média, 3.28 segundos, o que reforça ainda mais este indício.

Número de usuários que não conseguiram realizar a tarefa

Devido o sucesso de todos os participantes em concluir as tarefas propostas, não houve uma análise do número de usuários que não conseguiram realizar a tarefa.

Nível de dificuldade no uso do sistema

O nível de dificuldade foi medido a partir das respostas do questionário apresentado na Tabela 1. A Tabela 3 apresenta o resultado do questionário e traz a porcentagem de respostas para cada pergunta sendo a primeira coluna a tarefa apresentada na Tabela 1.

Tabela 3: Resultado do questionário.

-	1	2	3	4	5	6	7
1	0%	0%	12.5%	0%	12.5%	37.5%	37.5%
2	0%	0%	0%	12.5%	25%	37.5%	25%
3	0%	0%	25%	12.5%	12.5%	12.5%	37.5%
4	0%	0%	0%	12.5%	12.5%	37.5%	37.5%
5	0%	0%	0%	0%	0%	25%	75%
6	0%	0%	0%	37.5%	12.5%	37.5%	12.5%

Pela Tabela 3, é possível notar que a maioria dos usuários concordam totalmente que o aprendizado do sistema foi bastante fácil (linha 1) e que o mesmo pode aumentar a produtividade de suas atividades (linha 2). Quanto às mensagens de erro apresentadas pelo sistema, o nível de satisfação foi menor e esse é um ponto onde o ambiente poderá ser melhorado. Já as informações do sistema (linha 4) e sua interface (linha 5) foram bem avaliadas pelos participantes. Quanto as funções e capacidades do sistema (linha 6), há novamente um ponto onde o ambiente pode ser melhorado.

Problemas encontrados durante os testes

Os problemas reais que os participantes enfrentaram foram listados e classificados, conforme apresentado na Tabela 4.

Estes problemas encontrados serão solucionados pela equipe de desenvolvimento da ferramenta.

5. CONCLUSÃO

Este trabalho apresentou o Teste de Usabilidade do Mosaicode, avaliando a produtividade, a facilidade e eficiência de aprendizado e de uso e a satisfação do usuário.

Como resultado, foi apresentado análises do tempo de conclusão das tarefas e um questionário avaliando a satisfação de uso pelo usuário. Pela análise do tempo para execução de tarefas concluiu-se que os participantes conseguiriam otimizar cada vez mais o tempo de desenvolvimento de Tarefas de acordo com o tempo de utilização da ferramenta.

Pelo resultado do questionário pode-se concluir que o Mosaicode torna mais fácil e prático o desenvolvimento de aplicações para o seu domínio específico, atendendo aos requisitos de uma VPL.

Pelo fato deste ambiente gerar código baseado nos plugins utilizados pelo usuário, questões como portabilidade, confiabilidade, otimização, redução dos custos de manutenção e testabilidade do código são implícitos a maneira como a ferramenta trabalha. Para testar, dar manutenção ou otimizar o código gerado é necessário alterar os plugins do ambiente, tarefa essa que não foi avaliada neste trabalho.

Tabela 4: Problemas apontados pelos participantes.

Problemas	Classificação
Dificuldade para compreender as entradas/saídas dos blocos - para que servem e os seus tipos	Cosmético
Dificuldades para organizar/alinhar os blocos de forma que não haja sobreposição de conexões	Cosmético
Dificuldade para fechar a janela de vídeo na Tarefa 1	Cosmético
Os blocos desaparecem quando o zoom é muito minimizado	Cosmético
O programa fica lento ao clicar em *run* duas vezes consecutivas, na Tarefa 1. Continua lento mesmo depois de fechar as janelas	Catastrófico
Aparece uma mensagem de erro, sem nenhuma informação, ao clicar na opção de *salvar* e, em seguida, no X (para fechar)	Cosmético
Dificuldade para encontrar os plugins devido o fato dos mesmo não estar em ordem alfabética	Cosmético

Por fim, foi apresentado uma lista de problemas reais que os participantes enfrentaram, classificados por sua gravidade. A identificação e solução desses problemas permitirá melhorar a usabilidade do sistema.

Os autores agradecem a Universidade Federal de São João Del Rei pelo apoio financeiro a este projeto e ao professor Dárlinton Barbosa Feres Carvalho pela oportunidade de desenvolver este trabalho em sua disciplina.

6. REFERÊNCIAS

[1] S. Barbosa and B. Silva. *Interação Humano-Computador*. Elsevier Brasil, 2010.

[2] K. G. Ferreira, M. d. F. de Curso, and C. I. P. da Silva. Teste de usabilidade. *Trabalho de conclusão de curso. Universidade Federal de Minas Gerais. Belo Horizonte, MG*, 2002.

[3] A. Gomes, J. Henriques, and A. Mendes. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 1(1):93–103, 2008.

[4] P. E. Haeberli. Conman: A visual programming language for interactive graphics. *SIGGRAPH Comput. Graph.*, 22(4):103–111, June 1988.

[5] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

[6] A. C. T. Klock, I. A. Campos, I. Gasparini, and M. d. S. Hounsell. Avaliação de usabilidade de sistemas de gerenciamento de referências bibliográficas. 2016.

[7] A. Van Deursen and P. Klint. Domain-specific language design requires feature descriptions. *CIT. Journal of computing and information technology*, 10(1):1–17, 2002.