

Programação para Artistas

Bruno N. Santos Francisco C. C. Marinho* Marília L. Bergamo* Rosilane R. Mota**

Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Belo Horizonte, Minas Gerais, Brasil

*Universidade Federal de Minas Gerais, Escola de Belas Artes, Belo Horizonte, Minas Gerais, Brasil

**Pontifícia Universidade Católica de Minas Gerais, Instituto de Informática, Coord. Sistemas de Informação, Brasil

Resumo

Este artigo tem como objetivo, realizar uma comparação entre os ambientes de programação, Processing e OpenFrameworks. Assim como um estudo de caso de um jogo implementado nos dois ambientes. A comparação, se limita no escopo, do aprendizado, desenvolvimento de código, eficiência e utilização de diferentes interfaces.

Palavras-chave: open source artist processing open frameworks programação jogos

Contato:

bnsantos@dcc.ufmg.br
chicomar@uol.com.br
marilialb@yahoo.com
rosilane@pucminas.br

1. Introdução

O intuito deste documento é fomentar a discussão de algumas ferramentas open source utilizadas por artistas para produzir jogos e a chamada “*Algorithmic Art*” que é um tipo de arte gerada, composta ou construída por algoritmos. Este tipo de arte utiliza várias teorias matemáticas com o objetivo de produzir de forma de arte automatizada e randômica.



Figura 1: Exemplo de *Algorithmic Art* utilizando o conceito de Fractal;

As plataformas de desenvolvimento discutidas compartilham um objetivo, o de procurar facilitar a programação encapsulando alguns acessos de baixo nível a dados ou modificando a linguagem de programação tentando simplifica-la. Para aqueles que não sabem programar possam iniciar neste ramo de um modo simplificado.

A discussão terá enfoque nas plataformas de desenvolvimento Processing e openFrameworks, relevando seus pontos fortes e suas deficiências separadamente, assim como uma comparação entre as duas ferramentas ao final. A análise terá enfoque nas questões de aprendizado, facilidade de configuração, capacidade de adicionar recursos e o uso em diferentes interfaces. Na comparação entre as duas plataformas foi feito um estudo de caso, implementou-se um jogo nas duas plataformas, para comparar a implementação e a eficiência entre elas.

2. Processing

Processing é a primeira ferramenta criada para artistas por artistas, o seu desenvolvimento foi iniciado no [MIT Media Lab](#) por dois estudantes de graduação Casey Reas e Benjamin Fry em 2001. A meta inicial de seus criadores era encontrar uma forma de escrever código fonte que dê suporte e possa inspirar o processo de criação.

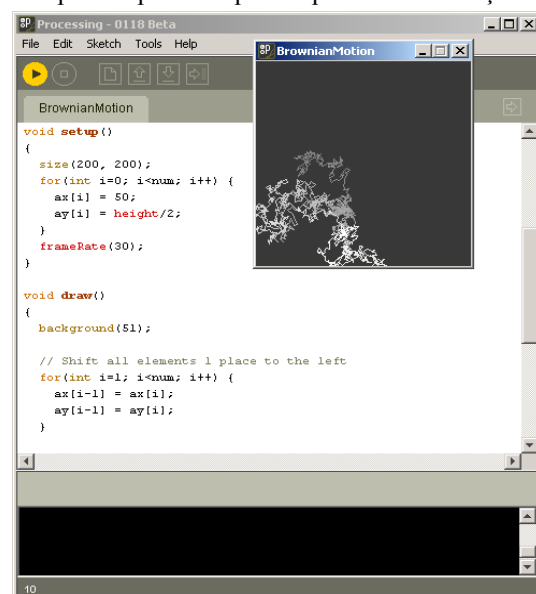


Figura 2: Screenshot da IDE do processing ;
Processing é uma linguagem de programação open source e um ambiente de desenvolvimento para alunos,

artistas, designers e pesquisadores programarem imagens, animações, jogos e interações. A especialidade da linguagem de programação Processing é gerar e modificar imagens.

O ambiente de desenvolvimento do Processing é bastante simples, ver figura 2, comparado a outros ambientes de desenvolvimento existentes no mercado. A simplicidade é uma forma de atrair usuário que não possui experiência em programação. Desta forma um iniciante pode criar um programa com poucos minutos de estudo, e usuários avançados podem fazer bibliotecas adicionais com maiores recursos e acrescentá-las ao Processing. Este suporte aos diferentes níveis é uma das metas do Processing.

Processing é uma linguagem parecida com JAVA, ou seja, Processing foi feito baseado em JAVA com certas simplificações e alguns acréscimos na linguagem. Possui licença GLP e LGPL. A GLP diz que todo trabalho derivado também deve ser open source, já a LGPL deixa livre a escolha da licença dos trabalhos derivados.

2.1 Vantagens

A principal característica do Processing é o fácil aprendizado, a modificação feita em sua linguagem base juntamente com os métodos de ensino disponíveis (exemplos, comunidades e livros texto) tornou o aprendizado extremamente rápido, bastando alguns minutos de leitura e poucas linhas de código para conseguir resultados incríveis.

Outro fator importante é a sua instalação, basta fazer o download da IDE compilada para a sua plataforma (Linux, Windows e MAC OS) que possui o compilador embutido e então executar a ferramenta. A IDE do Processing é bastante simplificada e amigável, ver figura 2. Permite gerenciar projetos simples. Possui um sistema, relativamente simples, de adição de novas bibliotecas, bastando adicionar a nova biblioteca no diretório de bibliotecas do Processing.

Possui uma comunidade muito grande, talvez pelo fato de ser uma das primeiras ferramentas do seu tipo. A sua comunidade possui um grande número de usuários ativos e dedicados. Além da comunidade no qual pode-se conseguir ajuda, diversos exemplos e novas bibliotecas feitas por entusiastas, Processing possui diversos livros texto que ensinam a utilizar o ambiente de desenvolvimento os quais alguns são citados nas referências. E principalmente o apoio do [MIT Media Lab](http://www.media.mit.edu/).

No quesito de utilização em diferentes interfaces o fato de utilizar como linguagem base JAVA faz com que Processing tenha várias bibliotecas para acessar dispositivos via bluetooth, porta serial, etc. E possui fácil comunicação com dispositivos como o Arduino que pode ser acessado em <http://hardware.processing.org/>. Um exemplo prático foi a completa comunicação entre o Processing e o wiimote que pode ser comprovada na figura 3, além deste exemplo outros serão citados no item 2.3.

2.2 Desvantagens

O fato da linguagem base ser JAVA, resulta em vantagens e desvantagens. A principal desvantagem está no quesito desempenho, pois JAVA é uma linguagem no qual o código final não é executado diretamente na máquina alvo e sim interpretado pela máquina virtual JAVA da plataforma vigente. Em aplicações que exigem o máximo de recursos gráficos, como jogos, isto é um fator que causará queda no desempenho, consequentemente atraso na exibição de uma cena o que leva o jogador a parar de jogar.

A IDE por ser simples não suporta a criação de projetos mais complexos. A criação de bibliotecas deve ser feita em outro ambiente de desenvolvimento.

2.3 Exemplos

A seguir serão apresentados alguns exemplos de programas e jogos feitos utilizando o Processing.

1)



Figura 3: Vídeo de uma aplicação que utiliza o Wiimote.

Fonte: <http://www.vimeo.com/1489202>

2)

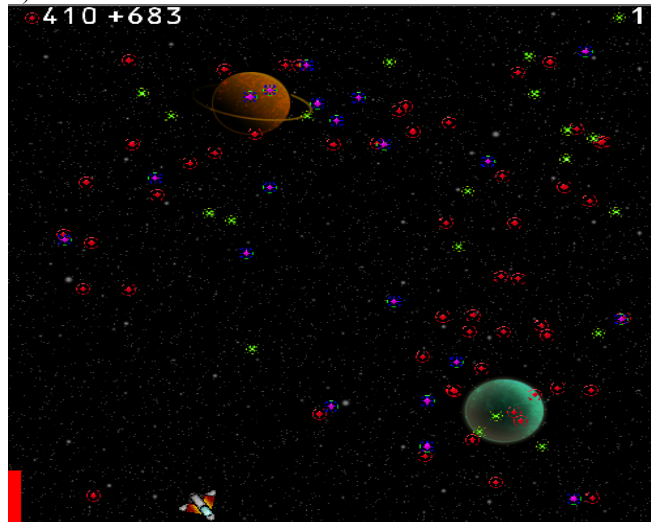


Figura 4:Jogo Space Rider feito em Processing.

Fonte:<http://space.wliia.org/>

3. Open frameworks

O OF(Open Frameworks) é desenvolvido e mantido por Zachary Lieberman e Theo Watson. Os dois possuem formação artística, o primeiro é americano com bacharelado no Hunter College em artes e um mestrado de

design e tecnologia em Parsons School of Design, já o segundo é britânico e possui bacharelado em arte e design na Parsons School of Design. Apesar de possuir basicamente os mesmos objetivos que o Processing o OF não possui uma IDE simples própria e a linguagem é o C++.

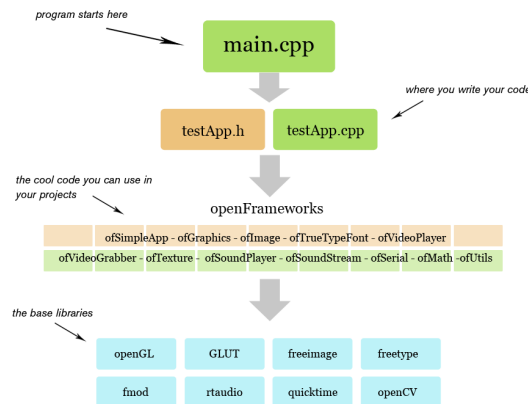


Figura 5: Estrutura básica de um programa em OF ;

OF não é uma linguagem de programação, ele é um framework. Um framework é uma ferramenta que fornece determinadas funções que provêm acesso a determinados dados de baixo nível, necessários para o correto funcionamento da aplicação, de forma encapsulada, ou seja o programador pode manipular estes dados em alto nível de abstração e até mesmo não manipular estes dados em alguns casos. Desta forma consegue-se programar em um nível de abstração ainda maior.

Uma analogia que pode ser feita com o conceito de framework é que você está dirigindo seu próprio filme e decidiu trabalhar com uma empresa produtora de filmes. A empresa arruma o local, fornece toda a infra-estrutura necessária. Ela faz com que todo mundo esteja no set de gravação fazendo seu trabalho e você decide o que vai acontecer no set de gravação quando todo mundo estiver pronto. OF é a empresa produtora de filmes que cuida de todo o processo administrativo e logístico, deixando você se concentrar no processo criativo.

A estrutura de um projeto em OF segue um padrão que está representado na figura 5. O programador deve ter em seu projeto pelo menos 3 arquivos, o main.cpp, o testApp.cpp e testApp.h. A função destes arquivos é a seguinte:

- main: criar o objeto e inicializar a execução do programa.
- testApp: Instanciar o objeto testApp, derivado do objeto padrão ofSimpleApp, e sobrescrever os métodos definidos para a classe ofSimpleApp.

OF possui licença LGPL. Este tipo de licença coloca restrições no próprio programa, mas deixa livre a escolha do tipo de licença dos trabalhos que estão relacionados com este programa.

3.1 Vantagens

OF não possui seu próprio compilador, logo

qualquer usuário avançado pode utilizar as chamadas de baixo nível assim como as de alto nível disponíveis pelo OF. Este fator deixa o programador utilizar todos os recursos da extensa linguagem C++, recursos como herança de classe o qual é impossibilitado de ser utilizado no Processing.

Utiliza IDEs avançadas para a realização de projetos, as IDEs utilizadas são Code::Blocks (Windows/Linux), DevC++(Windows) e Xcode (Mac OS). Estas IDEs permitem o uso de ferramentas avançadas.

Muito fácil adicionar novas bibliotecas ao OF, contanto com um manual de como fazer isso no próprio site e uma comunidade de entusiastas que sempre estão produzindo novas bibliotecas. Existem bibliotecas de visão computacional, comunicação bluetooth, manipulação de vídeo (quicktime), etc.

3.2 Desvantagens

A linguagem C++ é uma linguagem que pode ser considerada difícil, em relação a JAVA, o uso de ponteiros é considerado por muitos um excelente recurso computacional mas é bastante complexo de ser utilizado.

Comparado com o Processing o OF ainda está em sua fase inicial, pois é um projeto relativamente novo ainda está na versão de teste. Logo possui deficiências em sua documentação e em manuais introdutórios. Mas acredito que com o tempo esta comunidade consiga diminuir estas deficiências. Assim o aprendizado se torna complexo e demorado para iniciante.

O fato de utilizar IDEs avançadas pode deixar o iniciante um tanto quanto perdido, por causa das inúmeras ferramentas disponíveis. Mas quando se deseja fazer projetos mais complexos não existe outra opção.

3.3 Exemplos

A seguir serão apresentados alguns exemplos de jogos e arte feita utilizando o Open Frameworks.

1)



Figura 7: Projeto Drawn

-drawn : projeto que faz com que desenhos feitos a tinta

parecem ter ganhado vida. Ver figura 7
<http://thesystemis.com/drawnInstallation/>
 2)

-Audio Space: projeto no qual você pode deixar uma mensagem em algum lugar da sala e para que outra pessoa consiga escutar essa mensagem ela deve estar exatamente no mesmo lugar no qual você deixou a mensagem. Figura 8.

http://muonics.net/site_docs/work.php?id=15



Figura 8 Equipamento utilizado no projeto Audio Space;

OF, cerca de 815 linhas, ficou maior do que o código em Processing, cerca 600 linhas. E a complexidade de programação foi maior no OF, devido a alocação dinâmica e utilização de ponteiros, facilidades que a linguagem JAVA possui em relação ao C/C++.

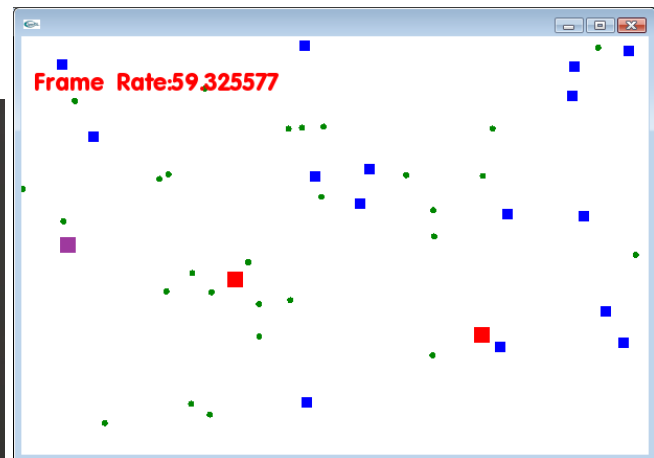


Figura 9: Screen do jogo rodando em OF.

No quesito eficiência, o código em OF se mostrou extremamente rápido em comparação ao código em Processing, utilizando tanto a biblioteca OpenGL e a P3D. Como pode ser comprovado no gráfico da figura 10.

4. Implementação- Processing vs OF

Como estudo de caso, implementou-se um jogo que utiliza como base a iteração entre diferentes tipos de espécies (Figura 9). No caso existem três tipos de espécie, as plantas, os herbívoros e os predadores. As plantas são representadas por círculos verdes, os herbívoros por quadrados azuis e os predadores por quadrados vermelhos. Na cadeia alimentar deste ambiente, os herbívoros podem se alimentar das plantas, mas são presas dos predadores. E cada espécie, exceto as plantas, podem se reproduzir com o contato direto entre outro indivíduo da mesma espécie. O jogador controla um predador, que é representado pela cor roxa de forma a diferenciá-lo dos outros predadores, que pode se movimentar por todo o plano utilizando as setas direcionais, para comer um herbívoro basta colidir com o mesmo e para a reprodução é necessário o contato direto com outro indivíduo da mesma espécie.

O objetivo do jogo é fazer com que os predadores salvem as plantas, dessa forma é necessário que se elimine a espécie dos herbívoros cuja reprodução é muito rápida, antes que eles acabem com as plantas.

O código em Processing, foi feito com base nos trabalhos de arte e vida artificial de Emiliano Causa, os códigos em Processing do Emiliano Causa estão disponíveis no site:

http://www.biopus.com.ar/emiliano/tutorial_vida_artificial/index.html

A implementação em OF foi feita de modo que ficasse o mais próxima possível da implementação em Processing, para compararmos, a dificuldade de programação, tamanho do código fonte e eficiência. Para comparação da eficiência é feita uma análise entre o número de organismos e o frame rate. O código fonte em

Processing vs OF

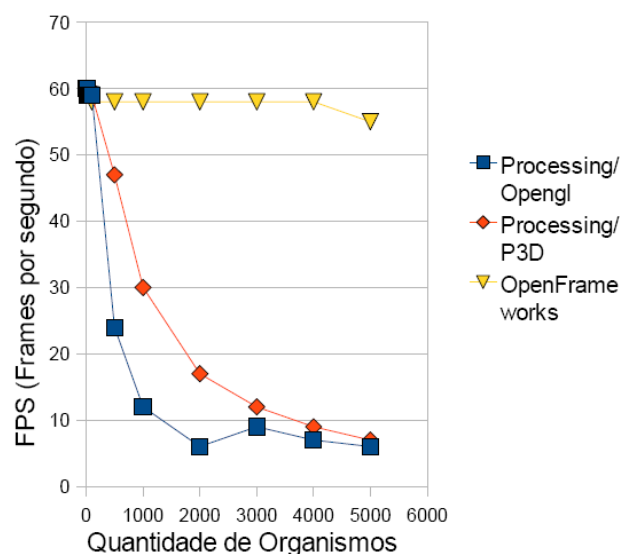


Figura 10: Gráfico comparativo entre o FPS do Processing e o OF.

Os resultados obtidos neste documento foram realizados num laptop DELL XPS M1530, Intel® Core™ 2 Duo Processor T7250 2,00GHz, 3GB Dual Channel DDR2 SDRAM at 667MHz, 160GB 7200rpm.

5. Conclusão

Até agora citamos alguns dos pontos fortes e deficiências de cada uma das duas ferramentas para artistas programarem. Apesar de todas as análises, nenhuma das ferramentas é melhor que a outra incondicionalmente, e sim, cada uma irá se encaixar perfeitamente de acordo com

a sua necessidade.

melhorados ate sua versão final.

Processing é uma ferramenta de fácil utilização e aprendizado, que apesar das suas deficiências, estas qualidades se sobressaem em relação ao OF. E por isso Processing é indicada para iniciantes em programação, pois jogos mais complexos, que necessitam da renderização de vários objetos, irão sofrer com a eficiência do Processing.

Referências

- REAS, CASEY AND FRY BEM, 2007. PROCESSING: A PROGRAMMING HANDBOOK FOR VISUAL DESIGNERS AND ARTISTS – MIT PRESS
- GREENBERG, IRA, 2007. PROCESSING: CREATIVE CODING AND COMPUTATIONAL ART – FRIENDS OF ED

Para aplicações que necessitam de uma resposta rápida, como jogos e aplicações de tempo real, o OF é a melhor opção. Apesar de ser considerada difícil em relação ao Processing, suas aplicações apresentaram maior eficiência em relação ao Processing. Logo para aplicações avançadas a melhor escolha seria OF.

A análise de qual ferramenta a ser utilizada depende de um tradeoff, entre eficiência e dificuldade de uso. Para aqueles que nunca programaram a melhor escolha seria o Processing, mas se você possui alguma prática de programação é aconselhável a utilização de OF, para obter aplicações com maior eficiência.

O OF está em sua versão beta, espera-se que muitos dos pontos fracos citados aqui possam ser