

Projeto de Iniciação Científica

Técnicas Avançadas de Programação
Multimídia Multiplataforma para Videogames
Unificando a Web, Java, e sistemas operacionais móveis

Prof. Ricardo Fabbri, Ph.D.



Grupo de Visualização
Instituto Politécnico – IPRJ
Universidade do Estado do Rio de Janeiro

Nova Friburgo, 15 de Março de 2013

Orientador

Prof. Ricardo Fabbri, Ph.D.

Projetos Relacionados

FAPERJ APQ1 01/09335-8

1 Contextualização

O presente plano de pesquisa se refere a atividades propostas a nível de iniciação científica, a serem realizadas no Laboratório de Visualização do Instituto Politécnico – IPRJ da UERJ, Departamento de Modelagem Computacional. Pretende-se trabalhar em conjunto com o grupo de pesquisa do Prof. Roberto Pinheiro e com outros pesquisadores na realização de projetos e publicação de artigos, tanto no IPRJ como no ICMC (Instituto de Ciências Matemáticas e de Computação) e IFSC (Instituto de Física) da USP de São Carlos, bem como a Brown University (EUA). Também pretende-se colaborar de perto com coletivos de desenvolvedores de software livre como o LabMacambira.sourceforge.net para a tecnologia de software e PulaPirata.com para o design de jogos, dentre outros.

2 Introdução e Justificativa

A computação atual compreende uma ampla diversidade de ambientes de software e hardware, coexistindo, por exemplo, computadores *desktop* e *smart phones* de diversas marcas, cada um com um sistema operacional diferente, ou mesmo com versões diferentes de um mesmo sistema. Mesmo dentro de uma mesma versão de sistema operacional existem diversas diferenças de bibliotecas instaladas e suas versões. Para minar a diferença entre esses sistemas e facilitar a programação, diversas abstrações foram propostas ao longo dos anos, dentre as quais:

- A linguagem C [1] e, mais recentemente, C++ [2], que assume um modelo simples e razoavelmente genérico de um computador.
- A linguagem Java [3].
- O Navegador Web em seu estado moderno com seus padrões HTML, Javascript e CSS.

As tecnologias acima estão, de modo geral, em ordem crescente de abstração, bem como em ordem cronológica da época em que foram adotados em ampla escala.



Figura 1: A interoperabilidade de diversos dispositivos é um dos grandes ideais almejados pela computação atual. (Fonte: [4])

As linguagens C/C++ [1, 2] permitem uma boa portabilidade de programas, desde que as bibliotecas utilizadas sejam também portáveis, além de ser necessária a recompilação e pequenos ajustes no código. As dificuldades de portabilidade geralmente decorrem de diversas diferenças oriundas de compiladores C (especialmente C++) em diferentes sistemas operacionais mas, mais fortemente, devido ao uso de bibliotecas dependentes do sistema operacional. Java e sua máquina virtual (bem como as linguagens interpretadas como Python [5]) são geralmente implementados em C, e facilitam o desenvolvimento multi-plataforma por fornecer um ambiente de *execução* (não apenas compilação) razoavelmente uniforme e independente do sistema operacional. Uma comunidade central de desenvolvedores se concentra em portar o código em C da máquina virtual ou interpretador para cada plataforma, e todo o software restante executa em uma camada acima, mais uniforme que o ambiente nativo. No entanto, nem todo dispositivo e sistema operacional de interesse geral contém uma máquina virtual Java facilmente disponível. O Navegador web constitui uma abstração ainda mais profunda que Java, sendo inclusive base de sistemas inteiros chamados Web OS [6], como Chrome OS [7] e Firefox OS [8]. De fato, o Navegador Web inclui (ou deveria incluir) Java como uma de suas tecnologias padrão. No entanto, o suporte a Java nos navegadores atuais não é bom, sendo um dos motivos uma famigerada disputa entre a Sun e Microsoft, fazendo com que esta última removesse a JVM (Java Virtual Machine) do popular navegador Internet Explorer.

Para superar problemas históricos com o Java, todo navegador moderno passou a suportar de fato linguagens de *script* como Javascript para programação multi-plataforma. No entanto, Javascript é uma linguagem sub-ótima para programação de aplicativos de grande escala, não tendo sido desenvolvida para tanto. Por outro lado, Java foi projetada desde o início para programação de software de grande porte, tendo sido aplicada com sucesso nesse nicho, além de ser razoavelmente portátil. Ademais, a tecnologia de navegadores modernos com Javascript também não permite uma portabilidade total, já que existem diferenças programáticas (intencionais ou não) entre navegadores diferentes, ou mesmo entre versões de um mesmo navegador, bem como entre um mesmo navegador em diferentes dispositivos e sistemas operacionais. Dessa forma, a programação moderna é tipicamente multi-navegador (*cross-browser*) além de ser multi-plataforma (*cross-platform*).

O presente projeto pretende fazer com que o aluno ganhe experiência com técnicas modernas de desenvolvimento de software multi-plataforma. Em particular, pretende-se estudar modos de se unir a força do Java e do Navegador Web, que avaliamos como sendo as mais modernas e mais utilizadas tecnologias multi-plataforma. Para explorar os limites dessas tecnologias, iremos abordar programação multimídia, incluindo audiovisual e interativa/tempo real. Apenas recentemente têm-se desenvolvido padrões Web para tratar de processamento de áudio, vídeo e interatividade de forma multi-plataforma e consistente.

Um dos aplicativos mais importantes na linha multimídia em tempo real são os *videogames*. Os jogos de computador geralmente exercitam diversas tecnologias simultaneamente: áudio, vídeo, interação, e processamento em tempo real. Ademais, um *videogame* é genericamente modelado como um simulador, sendo que a mesma tecnologia pode ser aplicada para o desenvolvimento de simuladores para qualquer sistema em geral, desde simulação de fluidos, aviões, navios, carros, esportes, até simulações de deposição de polímeros, simulações da mente humana, e assim por diante. Tradicionalmente, jogos têm exercitado tanto as diversas funcionalidades dos computadores, que eles têm sido uma das grandes forças motrizes por trás da lei de Moore, em especial na manutenção do crescimento exponencial do poder de processamento das modernas GPUs [9].

Outra motivação para Jogos é que os mesmos podem ser alavancados para mudança social. Uma tendência atual é construir jogos como simuladores de situações que possam ser mais diretamente úteis para a sociedade, como simuladores de hospitais, etc. Assim, pretendemos focar este projeto na programação de jogos que simulem uma situação real, com por exemplo um simulador de um estabelecimento de negócios ou um simulador do desenvolvimento da personalidade humana. A idéia é que além de permitir a exploração das tecnologias modernas e avançadas de programação, os *videogames* permitam também a criação de aplicativos que ajudem a treinar pessoas para alavancar uma mudança na sociedade ou ajudem uma maior conscientização de seus meios. Videogames ou simuladores são aplicativos interativos que constituem uma forma moderna de expressão humana, extendendo as possibilidades tradicionais dos filmes, livros, pintura, etc.

Algumas das tecnologias mais avançadas para programação multiplataforma têm sido desenvolvidas para jogos. Dentre elas, podemos citar a Google PlayN [4], que faz uso de uma engenharia de software bastante complexa de compilação e tradução de código para desenvolvimento verdadeiramente multiplataforma. A tecnologia PlayN consiste na escrita de uma base de código multi-plataforma em Java e sua posterior compilação e tradução para o código nativo de inúmeras plataformas: Java nativo, Java para Android, iOS, HTML5/Javascript, dentre outros, como na Figura 2.

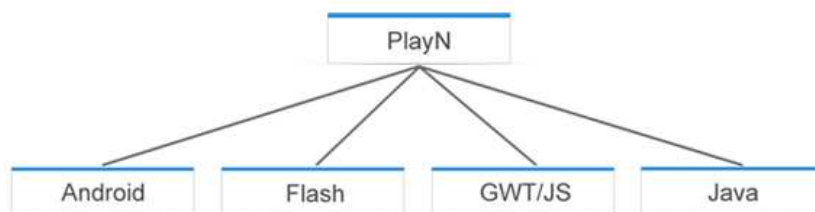


Figura 2: A tecnologia Google PlayN consiste na escrita de uma base de código multi-plataforma em Java e sua posterior compilação e tradução para o código nativo de inúmeras plataformas: Java nativo, Java para Android, iOS, HTML5/Javascript, dentre outros. (Fonte: [4])

A tecnologia PlayN é bastante recente e procura solucionar os problemas a serem estudados neste trabalho. No entanto, a tecnologia é bastante complexa e incipiente. Idealmente, ela deverá fornecer a interoperabilidade de diversos dispositivos, o que é uma das grandes ambições dos fundadores da empresa Google, Figura 1. Apesar da biblioteca ser feita para jogos, a mesma tecnologia pode ser aplicada à programação verdadeiramente multiplataforma em geral.

No caso especial da tradução do código Java para HTML5/Javascript, a biblioteca PlayN utiliza o sistema de compilação de uma outra tecnologia avançada do Google voltada para o desenvolvimento multi-browser, chamada Google Web Toolkit (GWT). Uma grande questão é se a tecnologia atual HTML5/Javascript já evoluiu o suficiente para ser uma solução multi-plataforma em si, sem necessidade de tradução de outras linguagens como Java no caso do PlayN/GWT. Nossa pesquisa preliminar aponta para o fato de que uma tecnologia de solução abrangente como o PlayN sempre será útil, uma vez que ambientes heterogêneos são sempre inevitáveis. Com este projeto, pretende-se investigar a magnitude da utilidade dessas tecnologias de tradução de código para desenvolvimento multi-plataforma no contexto atual de HTML5/Javascript cada vez mais bem padronizado.

3 Objetivos

O presente projeto pretende fazer com que o aluno ganhe experiência com técnicas modernas de desenvolvimento de software multi-plataforma, desenvolvendo um estudo comparativo e, possivelmente, propondo novas soluções ou *best practices*. Em particular, pretende-se estudar modos de se unir a força do Java e do Navegador Web, que avaliamos como sendo as mais modernas e mais utilizadas tecnologias multi-plataforma. Para explorar os limites dessas tecnologias, iremos abordar programação multimídia, incluindo audiovisual e interativa/tempo real. Apenas recentemente têm-se desenvolvido padrões Web para tratar de processamento de áudio, vídeo e interatividade de forma multi-plataforma e consistente.

Como explicado mais extensamente na Introdução, um dos aplicativos mais importantes no âmbito multimídia em tempo real são os *videogames*. Os jogos de computador geralmente exercitam diversas tecnologias simultaneamente: áudio, vídeo, interação, e processamento em tempo real. Como tais tecnologias têm sido padronizadas apenas recentemente em navegadores Web, vemos que *videogames* constituem uma boa classe de aplicativos para o teste de dos limites das tecnologias multi-plataforma atuais.

O objetivo concreto do aluno será explorar as tecnologias supracitadas para desenvolver *videogames* em software livre, almejando o funcionamento do aplicativo nas seguintes plataformas: Linux, Windows, Mac OS, Android, iOS, Navegadores Firefox, Chrome e Safari (desktop e dispositivos móveis). Mais especificamente, neste projeto deverá ser determinado se é mais adequado programar para tais ambientes apenas usando HTML5+Javascript, Java, ou algum tipo de combinação como Google Web Toolkit

(GWT) e Google PlayN. Pretende-se determinar qual tecnologia ou *framework* multi-plataforma é mais adequada, e suas vantagens/desvantagens. A princípio a comparação será focada em HTML5/Javascript versus Google PlayN, sendo essas as tecnologias mais promissoras de acordo com pesquisa preliminar realizada para este projeto envolvendo listas de discussão e programação de um aplicativo simples.

4 Metas

Elaboração de Publicações Como metas específica deste projeto, tem-se a elaboração de uma publicação em congresso ou revista contendo um estudo comparativo das diferentes técnicas modernas de programação multi-plataforma.

Programação de Software Como meta deste projeto necessária à realização do estudo comparativo, tem-se a programação de um software multimídia usando-se tecnologias multi-plataformas e com software aberto ou livre. A princípio a comparação será focada em (ao menos) HTML5/Javascript versus Google PlayN. O software completo deverá ser implementado em PlayN, sendo que HTML5 deverá ser aprendido de maneira secundária. Iremos escolher um *videogame* que funcionará em Web bem como em Java nativo e em navegadores para celular, sendo que o aluno irá documentar as dificuldades ao longo do processo de desenvolvimento.

5 Metodologia

A Metodologia está descrita separadamente nos planos de trabalho para cada aluno.

6 Conclusão

Este projeto da área de computação propõe a exploração e comparação das principais técnicas modernas de desenvolvimento avançado multi-plataforma, com foco em Java e *framework* PlayN/GWT do Google, versus HTML5/Javascript puro em navegadores web. Para tanto é adequado a programação de aplicativos multimídia, audiovisuais e em tempo real, os quais exploram os limites da tecnologia multi-plataforma atual. Alguns dos melhores representantes desses aplicativos são *videogames*, os quais podem ser considerados verdadeiros simuladores de sistemas dinâmicos. Dessa forma, a comparação das tecnologias multi-plataforma será realizada no âmbito de um *videogame* que será desenvolvido neste projeto colaborativamente na internet. Com isso, pretende-se abordar um dos grandes problemas da computação atual – como melhor programar para dispositivos móveis e ao mesmo tempo para computadores *desktop* com diferentes sistemas operacionais, diferentes versões e diferentes navegadores? Explorando-se esta

questão com um aplicativo real e de código livre, pretende-se obter idéias acerca do futuro da computação e da Web. No decorrer deste trabalho, surgirão idéias de como melhorar a tecnologia atual para maximizar o benefício futuro à computação como um todo.

Bolsas requisitadas. Para este projeto estamos requisitando duas bolsas de iniciação científica, já que o assunto engloba muitos fatores, linguagens e ambientes que constituem um ecossistema de tecnologia que pode ser melhor assimilado por dois alunos. Cada aluno irá focar em um aspecto diferente: um em PlayN/GWT, e o outro em Web Standards/HTML5/Javascript.

Referências

- [1] B. W. Kernighan and D. M. Ritchie, *The C programming language*. Prentice Hall, 1988. 1, 2
- [2] B. Stroustrup, *The C++ programming language*. Addison-Wesley Longman Publishing Co., Inc., 1997. 1, 2
- [3] K. Arnold, J. Gosling, and D. Holmes, *The Java programming language*, vol. 2. Addison-wesley Reading, MA, 2000. 1
- [4] G. W. T. Team, “PlayN – overview.” <https://developers.google.com/playn>. 1, 3
- [5] G. van Rossum and et. al., “Python programming language,” 1991. 2
- [6] A. Wright, “Ready for a Web OS?,” *Communications of the ACM*, vol. 52, no. 12, pp. 16–17, 2009. 2
- [7] S. Pichai and L. Upson, “Introducing the Google Chrome OS,” *The Official Google Blog*, 2009. 2
- [8] P. D. Scott, “The Firefox smartphone OS,” 2012. 2
- [9] D. B. Kirk and W. H. Wen-meí, *Programming massively parallel processors: a hands-on approach*. Morgan Kaufmann, 2010. 3
- [10] R. Fabbri, L. Estrozi, and L. Costa, “On Voronoi diagrams and medial axes,” *Journal of Mathematical Imaging and Vision*, vol. 17, pp. 27–40, jul 2002.