

A Experiência de Desenvolvimento de Jogos Educativos Utilizando Softwares Livres

Lucas Diego Gonçalves C. Bruno Nascimento Santos* Carla Viana Coscarelli**

Pontifícia Universidade Católica de Minas Gerais, Tecnologia em Jogos Digitais, Brasil

*Universidade Federal de Minas Gerais, Ciência da Computação, Brasil

** Universidade Federal de Minas Gerais, Faculdade de Letras, Brasil



Figura 1: Exemplo de jogo de aventura em plataformas desenvolvido no Platinum Arts Sandbox, software livre para criação de jogos

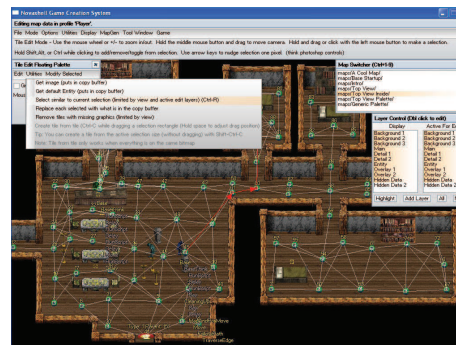


Figura 2: Criação de grafos de caminho utilizando a Novashell, uma ferramenta de autoria de código aberto para a criação de jogos 2D

Resumo

Esse trabalho pretende apresentar ferramentas gratuitas e de código aberto que possam ser utilizadas para o desenvolvimento de jogos. Exemplificando o uso e as funcionalidades desses softwares através da experiência do Projeto Aladim, um projeto de pesquisa e produção de jogos educativos para alfabetização de crianças. Buscamos através deste texto inspirar os desenvolvedores quanto à utilização de softwares livres e comprovar a possibilidade de desenvolvimento de jogos com utilização exclusiva de programas de código aberto.

Palavras-chave: software livre, código aberto, engines, ferramenta de autoria

Contatos dos autores:

lucasdgc@gmail.com

twitter.com/Lucasdgc

*bnsantos17@gmail.com

**cvcosc@yahoo.com.br

**twitter.com/cvcosc

1. Introdução

O Projeto Aladim, Alfabetização e Letramento em Ambiente Digitais Interativos Multimodais, é composto por uma equipe interdisciplinar de professores e alunos dos cursos de Belas Artes, Ciência da Computação, Jogos Digitais, Letras e Pedagogia UFMG e PUC-MG, e tem como objetivo principal criar jogos para auxiliar alfabetização e o letramento infantil.

Para isso o projeto trabalha com três frentes de pesquisa diferentes: o desenvolvimento de jogos para PC, o desenvolvimento de novas interfaces e a análise de jogos e sites educativos já existentes.

Com base no que já é oferecido no mercado em termo de jogos para alfabetização e letramento, percebemos produções que não conseguem aliar as habilidades pedagógicas à diversão e a desafios lúdicos, trazendo concepções *behavioristas*, sem um *feedback* rico o bastante para fazer o jogador compreender o erro e corrigi-lo em uma próxima tentativa [Ribeiro e Coscarelli, mimeo].

Em sua maioria, os jogos são remediações, ou seja, são atividades tradicionalmente feitas em papel e transportadas para a tela do computador, sem a exploração dos recursos tecnológicos disponíveis [Bolter 2001].

Os jogos existentes, quase em sua totalidade, não apresentam atrativos suficientes para prender a atenção dos jogadores por um longo período de tempo, não apresentam desafios variados, progressão gradativa da dificuldade [Rabin 2005] e nem trabalham habilidades pedagógicas diferenciadas.

A partir da experiência obtida no desenvolvimento de jogos, para o projeto Aladim, tenta aliar as habilidades pedagógicas necessárias no processo de alfabetização com a diversão, utilizando softwares livres procuramos fazer uma análise das ferramentas utilizadas no presente artigo.

Os jogos de alfabetização que estão sendo propostos por esse projeto têm como base uma concepção de aprendizagem na qual o jogador é visto como alguém que pode construir seu próprio conhecimento, podendo contar com o auxílio de instruções breves, feedback rico e com atividades que o levam a construir regras que regem o nosso sistema de escrita alfabético. Para isso, ele terá a oportunidade de, nos jogos, operar com elementos fônicos, alfabéticos, morfológicos, sintáticos e textuais, em tarefas que vão ajudá-lo a lidar tanto com os significantes quanto com os significados.

A iniciativa de utilizar softwares livres para o desenvolvimento dos jogos dentro do projeto veio junto com a necessidade de encontrar ferramentas completas o suficiente para desenvolvimento de aplicativos multiplataforma, da grande utilização de sistemas open source dentro das universidades, e problemas de compatibilidade de softwares proprietários em conjunto com sistemas operacionais Linux.

As ferramentas referidas a seguir foram utilizadas para o desenvolvimento de jogos, protótipos ou testes dentro do Projeto Aladim.

2. Ferramentas para desenvolvimento de jogos 2D

Para o desenvolvimento de jogos 2D já foram testadas diversas ferramentas, entre Kits de Desenvolvimento (SDK) a ferramentas de autoria. Todas as experiências relatadas são de programas já maduros, atualizados e multiplataformas. Os softwares foram avaliados quanto aos recursos e funcionalidades oferecidos, compatibilidade multiplataforma (Linux/Windows/Mac), documentação e suporte, e frequência de atualização (lançamento de novas versões) assim como data de lançamento da última versão disponível.

2.1 ClanLib



Figura 3: Logo da Clanlib

A ClanLib[ClanLib] é uma biblioteca para criação de aplicações 2D que utiliza C/C++. Possui suporte multiplataforma, podendo ser utilizado em Mac, Linux e Windows.

Experimentamos a ClanLib para criar um jogo de ação no estilo de *Space Invaders*, chamado Tiroletras. A ferramenta apresentou problemas de compatibilidade e integração com o Visual Studio Express 2008 e versões diferentes do Windows (Vista s/ SP, Vista SP1

e XP SP3). Em contrapartida foi facilmente compilada no Ubuntu 8.10.

A documentação da ClanLib é vasta e existe uma enorme gama de tutoriais no site oficial que explicam as funções mais básicas do funcionamento da biblioteca. O fórum apresenta um excelente suporte, por parte do tamanho da comunidade e por ter muitos usuários ativos, assim como possui diversos exemplos de soluções específicas para alguns jogos.

A biblioteca ainda está passando por melhorias lançadas periodicamente, porém sem intervalos fixos de tempo entre as versões. A mais recente é a 2.0.2, lançada em 21 de maio de 2009.

2.2 Box 2D

Box2D[Box2D] é uma biblioteca para simulação de física 2D. Já foi utilizada no desenvolvimento de muitos jogos comerciais como o recente Crayon Physics[Crayon Physics]. A biblioteca vem acompanhada com um editor, que permite colocar objetos na cena e testar os modelos de física disponíveis e as interações entre eles, como mostrado na Figura 5.

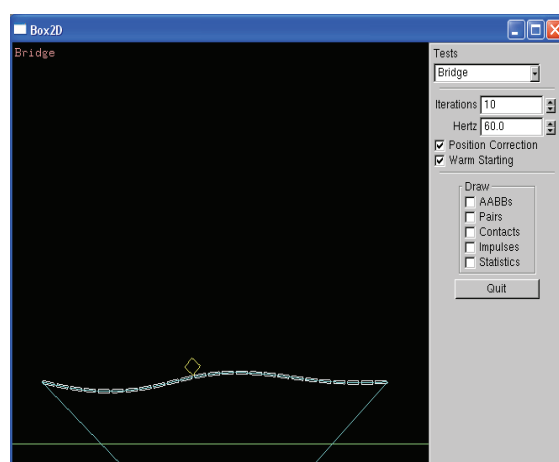


Figura 4: Simulação de cordas feita em Box2D

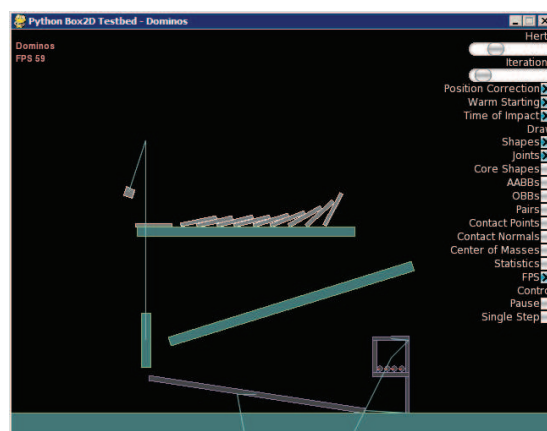


Figura 5: Visualizador de funções e modelos de física

A Box2D é capaz de realizar funções complexas pouco comuns em jogos bidimensionais de maneira

rápida e convincente. Entre as operações nativas na Box2D como exemplo são a tensão elástica, o empuxo e a fricção. Tudo pode ser agregado facilmente a qualquer projeto de jogo.

A biblioteca já vem inclusa em diversas engines, facilitando a implementação das funções do programa sem a necessidade da integração manual com o projeto, provendo a compatibilidade com diversos sistemas operacionais.

Possui uma documentação completa, com tutoriais básicos e avançados da utilização da biblioteca e um fórum satisfatório, já que a documentação é suficiente para esclarecer quase todas as questões relativas a Box2D.

Apesar de ser um programa maduro, ainda passa por atualizações, embora menos frequentes. A última versão (2.0) foi lançada em 13 de Abril de 2008.

2.3 Novashell

Novashell[Novashell] é uma ferramenta de autoria para jogos 2D. Funciona como um *Sandbox*, ou seja, é possível alternar entre jogar e editar apenas pressionando uma tecla. Possui um palco que pode ser dividido em uma grade com tamanho personalizável e os objetos instanciados podem ser posicionados visualmente nele, (ver WYSIWYG [Myers 1998]).

É possível também acessar uma janela de parâmetros do objeto através do editor gráfico, o que permite editar configurações específicas como tamanho, posição, atribuir um script Lua, referente ao funcionamento da instancia, e definir o modelo de física e colisão do objeto, feitos pela biblioteca Box2D que possui integração nativa com a Novashell.

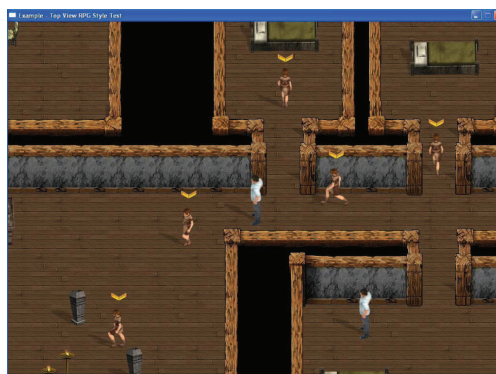
Existe também o suporte à criação grafos de caminho (waypoint graphs [Millington 2006]), que podem ser associados a qualquer instancia no palco como é mostrado na *Figura 2*.

A Novashell consegue gerar automaticamente sombras de sprites como mostrado na *Figura 6*, de acordo com a área de colisão aplicada sobre o sprite, podendo esta ser qualquer forma convexa. Esse é um recurso extremamente útil para jogos em que algum personagem é exposto a diferentes fontes de iluminação, descartando a necessidade da criação manual de todas as possibilidades de sombras.

Figura 6: Sombras geradas automaticamente em Novashell

O programa também possibilita a inserção de áudio dimensional na cena, apenas posicionando no palco o emissor do som, será calculado a intensidade do volume e balanço que será ouvido pelo jogador em cada ponto do cenário.

A ferramenta pode ser obtida para Windows, Linux



ou Mac, como executável ou código-fonte, para ser compilado. Como os jogos desenvolvidos na Novashell são desenvolvidos diretamente no editor gráfico, o projeto salvo funciona em qualquer sistema operacional compatível, sem a necessidade de recompilação.

Por ter todas as facilidades de uma ferramenta de autoria, a Novashell acaba se tornando burocrática e lenta para o desenvolvimento de determinadas aplicações que não envolvem os processos padrão previstos.

Utilizamos a ferramenta para o desenvolvimento do primeiro protótipo do Tiroletras, a ferramenta mostrou ser muito eficaz e rápida para a etapa de prototipação, mas pouco útil para a produção do jogo final.

O site oficial da Novashell conta com um conjunto de tutoriais que ensinam todas as funções realizáveis pela ferramenta e o fórum complementa a documentação, com o tratamento de dúvidas específicas para jogos desenvolvidos pela comunidade.

A ferramenta de autoria recebe atualizações a cada período entre 3 e 4 meses. Datando a última versão de 17 de maio de 2009.

2.4 Allegro

Allegro[Allegro] é uma biblioteca para a criação de jogos 2D/3D. Pode ser utilizada em Linux e Windows e é programável por C/C++. Apesar da versatilidade, é focada em programação de baixo nível, o que apesar de aumentar o tempo necessário pra implementação de funções básicas, agrega uma maior facilidade para transpor o projeto entre diferentes plataformas.

A biblioteca foi utilizada para a produção do jogo Tiroletras, devido à sua fácil portabilidade multiplataformas.

A documentação sobre a API é vasta o bastante para o desenvolvimento de projetos, contando ainda com um fórum com um bom número de usuários ativos.

Allegro possui uma atualização mensal desde o princípio deste ano. A versão atual é a 4.9.11, lançada em 7 de junho de 2009.

2.6 Love2D

Love2D[Love2D] é uma engine 2D que possibilita a criação de games através de scripts Lua. Possui uma baixa curva de aprendizado e rapidamente é possível implementar funções mais básicas.

A engine é focada em uma programação de alto nível, e facilita o desenvolvimento de jogos por grupos pequenos.

Também já possui as funções e recursos da Box2D implementados nativamente. A engine permite criar um arquivo único para a distribuição dos jogos com extensão própria (*.love), que contém todos os scripts, imagens e sons do jogo. Funciona semelhante ao .swf da Adobe, no qual é necessário apenas o arquivo para que o aplicativo seja executado.

A Love2D conta com um sistema de partículas leve e que permite diversas aplicações e customizações dos efeitos, conforme exemplificado na *Figura 7*.

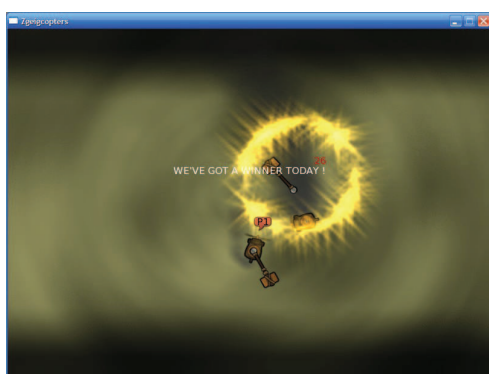


Figura 7: Efeitos gráficos procedurais na Love2D

A documentação da Love2D possui além de tutoriais e um forum, uma biblioteca de exemplos básicos feitos com a engine. Trata-se de um arquivo .love que contém exemplos em jogos das funções mais essenciais no desenvolvimento de aplicativos, como por exemplo colisão, gravidade, animação e controles de teclado e mouse.

O desenvolvimento de atualizações da Love2D está parado desde 28 de outubro de 2008, quando foi lançada a versão 0.5.

3. Ferramentas para desenvolvimento de jogos 3D

Desenvolver jogos em 3D com equipes pequenas é uma tarefa muito mais árdua e lenta comparada com os jogos bidimensionais. Isso acontece devido principalmente ao aumento da complexidade matemática, gerenciamento de um número maior de

processos simultâneos e dificuldades na otimização do desempenho do jogo para utilização em computadores menos potentes.

Com o intuito de desenvolver jogos tridimensionais foram experimentadas muitas ferramentas, à maioria destas se tratavam de engines cujo aprendizado era lento e de implementação vagarosa, o que acabou inviabilizando seu uso por questões de produtividade.

Os programas analisados adiante são aqueles que oferecem a possibilidade de desenvolvimento de jogos em 3D em conjunto com facilidades que otimizem o tempo de implementação. Existem muitas opções de engines e ferramentas de autoria gratuitas e de código aberto para jogos 3D, mas poucas dispõem de recursos que visam aprimorar o trabalho em grupos pequenos.

3.1 Blender game engine

Blender é um programa de modelagem tridimensional bastante difundido entre a comunidade artística. É um software muito versátil, capaz de realizar, além das operações básicas comuns em aplicativos do gênero, simulações de física, composição digital, criação de jogos, entre outras funções.

O Blender possui um editor de lógica chamado Blender Game Engine. Esse recurso faz com que os objetos presentes na cena sejam controlados e possam interagir com o usuário ou com outros objetos. A Blender Game Engine funciona como um sandbox, sendo possível fazer qualquer alteração na cena visualmente, sem a necessidade de recompilar o projeto.

O sistema de colisão e física do programa é feito pela *Bullet Physics Library*, uma biblioteca de código aberto, tornando possível a simulação de fluidos e corpos macios em tempo real.

A Blender Game Engine trabalha com scripts Python e SDL para gerenciamento de áudio.

Os aplicativos desenvolvidos com a engine são compatíveis com sistemas operacionais Windows, Linux e Mac, sem a necessidade de recompilação do projeto.

Blender é um dos softwares open source mais bem documentados, com uma enormidade de tutoriais, forums, vídeo-aulas e exemplos criados pela comunidade.

Utilizamos a Blender Game Engine para criar um protótipo 3D do jogo Papa-Letras, a ferramenta suporta bem os recursos gráficos do jogo, e responde bem aos controles, porém tem um desempenho muito baixo em computadores sem placa de vídeo dedicada.

A ferramenta passa por frequentes atualizações, com lançamento de novas funções e *plug-ins*. A versão mais recente é a 2.49, datada de junho de 2009.

3.2 Platinum arts sandbox

Platinum Arts Sandbox é uma ferramenta de autoria para jogos 3D construída sobre a engine Cube 2. A engine se caracteriza principalmente por unir recursos gráficos poderosos com a utilização de Shaders 2.0, alto desempenho computacional, funcionando estavelmente à 60 quadros por segundo em computadores medianos, e fácil criação de mapas e modificações.

Platinum Arts Sandbox é uma dessas modificações feitas, que eleva a facilidade de uso e produção da Cube 2, mantendo o desempenho e adicionando alguns recursos. O objetivo dos desenvolvedores é criar uma ferramenta simples o bastante para ser utilizada por crianças mas completa o suficiente para a produção de games completos.

A ferramenta permite a edição dos cenários em tempo real, criando manualmente relevos ou adicionando mapas de peso, sendo tudo alterado no próprio editor e não necessitando recompilação.

A importação de modelos e texturas para a Platinum Arts é feita adicionando os conteúdos desejados aos diretórios padrão da ferramenta, tornando-os assim disponíveis para edição e inserção no jogo. Todos os modelos importados podem ser transladados, rotacionados e escalonados diretamente no sandbox.

A Platinum Arts também traz um suporte à textos e diálogos, que podem ser exibidos mediante um evento padrão, como chegar em determinada parte do cenário ou conversar com algum personagem. Cabendo ainda a possibilidade de permitir que o jogador opte por diferentes opções de resposta.

Outro recurso interessante da ferramenta é a implementação de partículas, também inseridas através do editor gráfico. Além de fogo e efeitos genéricos, existe uma espécie de grama, que se posiciona perpendicularmente ao solo.

Desta maneira também é possível adicionar alguns tipos de materiais pré-definidos, entre eles vidro e água, que se comportam de maneira satisfatória, com as devidas reflexões e transparências.

Um dos aspectos que mais contribuem para o desempenho da Cube 2 é o gerenciamento de iluminação. Um comando calcula a intensidade luminosa em todos os pontos de todos os objetos na cena e essas informações são transformadas em mapas de luz (*Lightmaps*), que armazenam dados de brilho e coloração em consequência das fontes de luz. Isso faz

com que a engine não necessite de executar algoritmos de iluminação em tempo real; apenas acessar as informações dos *lightmaps* e integrá-las com as informações das texturas dos modelos.

Devido a proposta de ser uma ferramenta acessível a pessoas de qualquer idade, já estão inclusos alguns modos de jogo e mapas prontos, disponíveis com modelos já nativos da Platinum Arts. Entre os modos de jogo, que já trazem implementados os controles e funções mais comuns estão: RPG em primeira e terceira pessoa, Plataforma 3D com visão lateral, conforme exemplificado na *Figura 1* e *Save the Princess*, um modo de jogo onde o jogador tem como objetivo atravessar um cenário para alcançar a princesa.

A edição dos mapas e cenários do programa pode ser feita cooperativamente em rede LAN, onde até oito pessoas podem editar um mesmo mapa ao mesmo tempo.

A ferramenta foi compilada e executada em Linux e Windows, e os mapas salvos funcionam em todos os sistemas operacionais compatíveis com a Platinum Arts Sandbox.

A documentação da Platinum Arts Sandbox é suficiente para entender as funções e o funcionamento do editor, mas é necessário consultar a documentação da Cube 2 caso seja necessário um trabalho de codificação mais complexo. A comunidade presente no forum oficial da Platinum Arts assiste qualquer dúvida sobre a ferramenta, embora uma das regras da comunidade seja criar apenas conteúdo gratuito e de classificação livre.

A última versão da Platinum Arts Sandbox é a 2.3, lançada em Abril de 2009.

3. Considerações finais

A experiência obtida com a utilização de softwares livres na produção de jogos para o projeto Aladim prova a possibilidade de desenvolvimento utilizando apenas ferramentas de código aberto, obtendo inclusive diversas vantagens sobre o emprego de programas proprietários, principalmente mediante à necessidade de se portar o game entre diferentes plataformas.

Outra característica que pode contribuir pela opção por softwares livres é a vasta documentação disponível e o constante apoio da comunidade de usuários e desenvolvedores dos programas.

Porém, o mais importante aspecto das ferramentas de código aberto é a flexibilidade. O acesso ao código fonte e licença para alterá-lo provê infinitas possibilidades quanto ao desenvolvimento de jogos,

uma vez que não existem restrições do programa que não possam ser supridas com a manutenção do código.

Além de diversas opções de engines, bibliotecas e ferramentas de autoria de código aberto para a produção de jogos, também estão disponíveis outros softwares livres que realizam funções essenciais no desenvolvimento de jogos, como por exemplo edição de imagens e modelagem tridimensional.

Referências

BOLTER, J. David. *Writing Space: Computers, Hypertext, and the Remediation of Print*, Mahwah, New Jersey, Lawrence Erlbaum Associates Publishers, 2001

RABIN, Steve. *Introduction to game development*. USA: Charles River Media, 2005.

MYERS, Brad. *A Brief History of Human Computer Interaction Technology*. ACM interactions. Vol. 5, no. 2, March, 1998. pp. 44-54.

MILLINGTON, Ian. *Artificial intelligence for games*. USA: Morgan Kaufmann, 2006.

RIBEIRO, Andréa, COSCARELLI, Carla V. Jogos online para alfabetização: o que a Internet oferece hoje. Belo Horizonte: FALE/UFMG. (mimeo).

Sites oficiais dos programas apresentados:

Crayon Physics Disponível em:
<http://www.crayonphysics.com/>

ClanLib Disponível em: <http://clanlib.org/>

Box2D Disponível em: <http://www.box2d.org/>

Novashell Disponível em:
<http://www.rtsoft.com/novashell/>

Allegro Disponível em: <http://alleg.sourceforge.net>

Pygame Disponível em: <http://www.pygame.org>

Python Disponível em: <http://www.python.org/>

Love2D Disponível em: <http://love2d.org/>

Blender Disponível em: <http://www.blender.org/>

Platinum Arts Sandbox Disponível em:
<http://sandboxgamemaker.com/>