

Produção de Jogos Digitais ao alcance de todos: a criação de uma ferramenta para web 2.0

Hebert Phillipe Martins Pereira¹

Marcelo La Carretta²

Pontifícia Universidade Católica de Minas Gerais, Brasil.

RESUMO

O desenvolvimento de jogos é uma tarefa interdisciplinar que envolve assuntos como inteligência artificial, animação de personagens, física e som, sendo necessário o uso de uma ferramenta capaz de abstrair conceitos complexos e que possa gerenciar os diferentes tipos de arquivos utilizados. Um motor de jogos geralmente automatiza funções específicas do desenvolvimento, como carregar e exibir imagens, controlar a entrada de informações através de controles de jogo como teclado e joysticks e simular efeitos baseados em física. Porém, esse próprio motor costuma ser limitado à estação de trabalho na qual ele foi instalado, fora outros problemas como restrições de hardware e a nem sempre amigável interface dos motores de jogos atuais. O objetivo desse artigo é apresentar o diário de bordo do desenvolvimento de um motor de jogos baseado na web. Este motor pode ser executado em todas as plataformas que suportam HTML5, mantendo uma interface amigável e a expectativa é que ela possa ser mais uma ferramenta a ser utilizada na democratização da produção de jogos digitais.

Palavras-chave: game engine, web 2.0, prosumidor, edutech, mercado indie, html5.

1 INTRODUÇÃO

Desde o lançamento de Spacewar em 1962 [1] até o final da década de 80, o desenvolvimento de jogos digitais se dava por pequenos grupos de desenvolvedores, se não por uma única pessoa. Com o crescimento da indústria veio também o aumento da complexidade dos projetos de jogos digitais, tornando a produção de jogos uma tarefa demasiadamente longa e complexa, exigindo o envolvimento de pessoas com as mais diversas habilidades. Por esse motivo, as empresas passaram a necessitar de ferramentas que possam otimizar o desenvolvimento e possibilitar a entrega de novos jogos de forma rápida, mantendo a qualidade do produto. Segundo Gregory, o termo motor de jogos é usualmente utilizado para denominar um software que é extensível e pode ser usado como base para diferentes jogos sem grandes modificações [2]. Lewis e Jacobson completam dizendo que um motor de jogos é uma coleção de códigos que não especificam diretamente como será o comportamento ou a aparência do jogo. Esses códigos costumam ser responsáveis por funções como atualização dos estados do jogo, desenho, som e física genérica [3].

Motores de jogos são normalmente específicos de gênero. Contudo, há também uma grande quantidade de sobreposição tecnológica entre gêneros, especialmente no contexto de uma única plataforma. Gregory atribui isso ao advento de hardwares mais poderosos, fazendo com que as diferenças que existiam entre gêneros por causa da necessidade de otimização comecem a acabar, o que tem possibilitado o reuso do mesmo motor em

diferentes jogos [2]. Entretanto, a utilização de um mesmo motor em várias plataformas continua um desafio a ser vencido, devendo ainda serem feitas otimizações para garantir sua execução em outras plataformas além da qual o motor foi inicialmente projetado para. Portanto torna-se essencial o uso de uma abordagem que possa eliminar ou minimizar esses problemas.

A web nos últimos anos tem passado por uma revolução. Os usuários, antes meros telespectadores, agora possuem também a possibilidade de serem produtores, criando e alterando o conteúdo disponibilizado na web. Serviços como blogs, uma ferramenta que permite que os usuários compartilhem suas ideias na web sem a necessidade de conhecer sobre desenvolvimento web, serviços de vídeo como o YouTube e a Wikipedia, uma enciclopédia online que possibilita que qualquer usuário possa criar e editar os verbetes presentes no sistema, são bons exemplos do atual cenário da web, conhecida agora como web 2.0.

Segundo O'Reilly [4], um dos princípios da web 2.0 é trabalhar a web como plataforma, provendo serviços online que antes só poderiam ser executados se instalados em um computador, oferecendo também ferramentas que possibilitem aos usuários utilizarem o sistema de forma colaborativa, criando e editando o conteúdo e auxiliando no desenvolvimento do sistema como um todo, agindo não só como co-autores, mas também como co-desenvolvedores, uma característica do chamado prosumidor (consumidor e produtor) [5] e parte importante da composição de um cada vez maior mercado indie. O'Reilly também destaca que, na web 2.0, o serviço tende a se tornar melhor na medida em que mais pessoas o utilizam.

Se propõe, no projeto de pesquisa enunciado a seguir e explicitado no presente artigo, testar pontos onde a web pode ser usada como plataforma para um motor de jogos 2D, na tentativa de minimizar os desafios já citados. Para isso, será desenvolvido um motor de jogos baseado na web, visando seu uso em diferentes ambientes, sem a necessidade de instalação ou configuração dos mesmos. Ao motor gerado através da pesquisa foi dado o nome de Webty, junção das palavras web e liberty (liberdade).



Figura 1: Logo do motor de jogos abordado nesse artigo

¹ hebert.pereira@sga.pucminas.br

² marcelo@lacarretta.com.br

2 IMPLEMENTAÇÃO

Nessa seção, serão descritos o modelo de negócios e as funcionalidades já implementadas no motor Webty, assim como os conceitos básicos e a arquitetura do mesmo. O foco da pesquisa também é nos desdobramentos, buscando aprimorar esta ferramenta para que atenda satisfatoriamente o usuário que é produtor/consumidor, e fortalecendo assim o mercado de produções indie.

2.1 Conceitos básicos do modelo de negócios a ser adotado

Tomando como base os modelos citados por O'Really [4], foi definido um modelo de negócios a ser adotado pela Webty, tratando a ferramenta como um serviço e mantendo um relacionamento de codesenvolvedores com seus usuários. Serão adotadas também práticas de monitoramento dos usuários, visando identificar pontos fortes e fracos na Webty, permitindo que alterações possam ser feitas a curto prazo.

Como modelo de receita será adotado o modelo de assinatura descrito por M. A. Rappa [6], mesclando conteúdos pagos e gratuitos. A princípio, apenas os módulos que convertem o jogo criado para as plataformas Android e iOS e o módulo que auxilia na inserção de anúncios exigirão o pagamento de uma assinatura para serem utilizados.

2.2 Público Alvo

Com o intuito de demarcar melhor o objetivo a ser atingindo ao final do projeto, foi definido um público alvo, sendo esse pessoas interessadas em desenvolver jogos mas que não possuem conhecimentos em programação de computadores.

Tendo o público alvo em mente, deverão ser criados meios que possibilitem a esse público atingir o seu propósito, mantendo sempre a ferramenta alinhada as suas necessidades. Numa etapa futura, serão realizados testes com os usuários visando mensurar resultados e evidenciar problemas.

2.3 Conceitos Básicos da programação e do desenvolvimento

Antes de iniciar o desenvolvimento da Webty, uma regra foi estabelecida: Todo o projeto deve ser feito com o foco em dispositivos móveis, ou seja, visando atingir uma boa performance em dispositivos portáteis como smartphones e tablets. Essa regra foi definida por se acreditar que, caso esse objetivo fosse atingido, não haveriam problemas em propagar esse desempenho para dispositivos com maior capacidade de hardware.

A Webty está sendo desenvolvida usando tecnologias web como a linguagem de marcação HTML5 e a linguagem de programação Javascript. Arjan Egges define o desenvolvimento com HTML5 e Javascript como uma das poucas abordagens realmente democrática, funcionando em uma variedade de dispositivos, indo de computadores a smartphones e tablets [7]. Além disso, O'Reilly considera o Javascript como uma das tecnologias que ajudaram a revelar o potencial da web como plataforma para aplicações completas [2].

A Webty trabalha com o conceito de Worlds. Para começar a inserir entidades no jogo, você precisa primeiro ativar um World. Entidades são objetos no jogo que o jogador consegue manipular, modificar ou interagir de alguma forma [8]. Essas entidades podem ser adicionadas e/ou removidas pelo World. O motor suporta 5 tipos de entidades: Entity, Spritemap, Button, Text e Stamp.

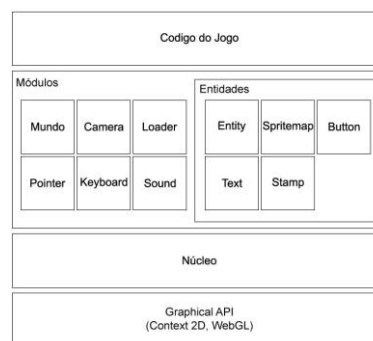


Figura 2: Arquitetura atual do motor

Nesse trabalho foi adotada uma estrutura modular para o desenvolvimento da Webty como a descrita por Lewis e Jacobson [3]. Os módulos da Webty são independentes, sendo obrigatórios apenas dois módulos para seu funcionamento: Núcleo e Câmera. Todos os outros módulos são opcionais. A seguir serão descritos os principais módulos já funcionais:

2.3.1 Núcleo

O Núcleo é o coração da Webty. Nele são feitas todas as configurações básicas, como tamanho e orientação (vertical/horizontal) da tela de jogo, além da taxa de quadros por segundo. Fica também à cargo do Núcleo as funções de atualizar, desenhar e redimensionar a tela, além de funções que auxiliam na localização do jogo. Localização é o processo de se traduzir o jogo para diferentes linguagens.

Segundo Smith, uma das grandes dificuldades do desenvolvimento para dispositivos móveis é a vasta quantidade de resoluções e tamanhos físicos existentes. O autor ainda completa dizendo que para um jogo ter sucesso nessas plataformas ele precisa ser capaz de se adaptar e tirar vantagem da grande quantidade de resoluções disponíveis para que os usuários possam experimentar o jogo na maior quantidade de dispositivos possível [9]. A Webty auxilia nessa tarefa ficando responsável por reconhecer o dispositivo no qual o jogo está sendo jogado e redimensionar a tela dinamicamente, de acordo com a resolução atual, tentando atingir o aspecto desejado. Para isso basta o desenvolvedor ativar essa funcionalidade nas configurações do motor. Smith se refere a essa habilidade como design responsivo.

2.3.2 World

World é onde o desenvolvedor consegue adicionar e/ou remover as entidades. Apenas um World pode ser ativado de cada vez, mas não há limite para a quantidade de Worlds criados. Sempre que um World for ativado, o anterior é destruído, eliminando também todas as entidades que haviam sido inseridas nele.

2.3.3 Câmera

A câmera é responsável por tornar visível o que está no World. Para economizar processamento, apenas o que está à frente dela é desenhado e atualizado (caso seja necessário, essa função pode ser desligada nas configurações da câmera). A Webty atualmente suporta apenas uma câmera.

Outras funções também já estão implementadas, como ajuste da intensidade e cor da luz da câmera e os efeitos shake, que faz a câmera tremer, e espelhar, que permite que a imagem sendo mostrada pela câmera seja espelhada em um ou dois eixos.

2.3.4 Entidades

Entidade é qualquer objeto que possa ser inserido no World. No momento o motor suporta cinco tipos:

- Entity é o tipo entidade padrão. sendo qualquer coisa que possua uma posição e que possa sofrer ou não colisão;
- Spritemap possui as mesmas características de Entity, além de poder ou não ser animado;
- Button é um tipo de entidade que possui interação por toque e/ou clique do mouse;
- Text é usado quando se deseja escrever algum texto na tela;
- Stamp é usado para se inserir uma imagem estática. Stamp possui apenas posição.

2.3.5 Módulos de entrada

Dois módulos de entrada são suportados: Keyboard e Pointer.

Keyboard permite a interação por teclado, podendo ser capturado o pressionar e o soltar de teclas e identificar qual tecla foi pressionada.

Pointer permite a interação por mouse e toque. Pointer identifica o tipo de entrada dinamicamente e se o usuário pressionou ou soltou a tela/o mouse. No caso de dispositivos de toque, até dez toques simultâneos podem ser identificados.

2.3.6 Loader

Loader é utilizado para carregar arquivos externos que deverão ser utilizados no jogo. Podem ser carregados sons e imagens.

2.3.7 Sound

Em jogos, sons são importantes por várias razões. Eles podem dar pistas para o jogador que algo está acontecendo, como o som de passos indicando que alguém se aproxima, ou que um botão foi pressionado. Arjan Egges lembra a importância do som como forma de aumentar a imersão do jogador, fazendo o ambiente mais vivo, mesmo quando nada está aparentemente acontecendo [7].

Van der Spuy cita duas formas de se executar sons em Javascript: usando o elemento de áudio do HTML5 ou a API Web Audio. Segundo o autor, a segunda opção é a mais adequada para jogos, por garantir o carregamento do som, permitir que vários sons sejam executados ao mesmo tempo e dar um controle mais preciso sobre o som que está sendo executado [10]. Apesar de todas essas vantagens, Bura e Coates ressaltam o fato de que a API Web Audio ainda não é suportada por todos os navegadores [11].

A Webty soluciona esse problema testando o suporte das duas opções no momento em que o jogo é inicializado e definindo o uso da alternativa suportada, priorizando sempre a API Web Audio. Isso garante que esse teste só será feito uma única vez, evitando que seja realizado um novo teste sempre que um som for carregado.

2.3.8 Tilemap

Bura e Coates definem Tilemap como uma poderosa técnica que permite construir grandes e complexas fases de jogos usando pequenas imagens chamadas de tiles [11]. Tiles são pequenas partes de uma grande imagem que podem se conectar entre si sem que haja uma quebra no novo padrão criado. A Webty auxilia na criação desse novo padrão oferecendo um módulo que possibilita a utilização dessa técnica de forma simples, bastando que o desenvolvedor forneça a imagem e um arquivo no formato .json contendo as orientações de como ele quer que a Webty crie esse novo padrão. Também é possível que alguns tiles sejam definidos como sólidos, fazendo com que eles colidam com outras entidades preestabelecidas pelo desenvolvedor.

3 PRIMEIROS TESTES

Como forma não só de averiguar a capacidade da Webty de se tornar um produto comercial, mas também de testar o seu potencial de construir produtos, foi desenvolvido com ela um jogo chamado Pimi Jumpers [12]. Esse jogo deveria utilizar todos os módulos já criados para a Webty, além de ser capaz de ser jogado tanto em dispositivos móveis, quanto em computadores pessoais.

3.1 Sobre o jogo

Pimi Jumpers é um jogo do gênero plataforma onde você controla os Pimis, seres felizes que amam pular para todo lado. O mundo deles está em perigo desde que monstros malvados o invadiram. Agora eles precisam da sua ajuda para acabar com esse problema terrível.



Figura 3: Tela Inicial do Jogo Pimi Jumpers

O jogo possui as seguintes características:

- 60 missões;
- 30 fases com 4 estilos diferentes (normal, espinhos, água e lava);
- 15 inimigos;
- 8 personagens (alguns com habilidades únicas);
- Sistema de localização com 4 idiomas suportados (inglês, português brasileiro, espanhol, italiano)

3.2 Diferencial

Como forma de diferenciar o jogo, foi criada uma mecânica na qual a tela do jogo se inverte de forma aleatória caso o jogador interaja com alguns tipos de monstros. A Figura 4 mostra esse efeito em prática.

A tela 1 mostra o jogo antes do efeito ser ativado, já a tela 2 mostra o jogo após o inimigo caracol rosa ter sido derrotado e como consequência a tela ter sido invertida horizontalmente. Esse efeito dura até que um inimigo caracol azul seja derrotado.

Graças a Webty, esse efeito pode ser criado sem grandes dificuldades, o que permitiu que o tempo economizado fosse investido em outras tarefas, como o desenvolvimento de novos inimigos e/ou novas fases. O jogo também conta com outras mecânicas interessantes, como teleporte, áreas com itens secretos, imã para atrair moedas e escudo para deixar o personagem invencível por um período de tempo.

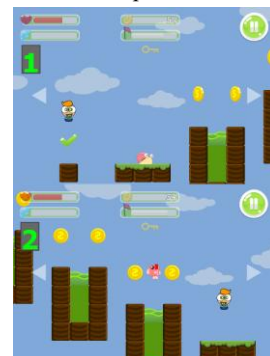


Figura 4: Tela do jogo antes e após o efeito de inversão

3.3 Recepção

Pimi Jumpers obteve um sucesso considerável para um jogo teste, tendo a sua licença de publicação adquirida pela Spilgames, uma das maiores companhias de jogos casuais do mundo. Além disso, Pimi Jumpers também foi vencedor do *Hoowu HTML5 Game Contest*, na China, na categoria inovação [13].

3.4 Desenvolvimento

Levando cerca de 25 dias para ser desenvolvido, Pimi Jumpers utilizou todos os módulos já funcionais da Webty citados anteriormente, fazendo com que fossem notados pontos que precisam ser melhorados para que a ferramenta cumpra melhor o seu objetivo.

Durante o desenvolvimento do jogo, surgiu a ideia de se utilizar efeitos de partículas em algumas situações. Por não ter um módulo de partículas já funcional, essa ideia teve que ser abandonada, sendo substituída por alguns efeitos de imagem, como rotação e escalonamento.

Outro problema encontrado foi a falta de uma interface capaz auxiliar no processo de desenvolvimento, removendo a necessidade de uso de ferramentas externas da Webty. Um exemplo disso é o desenho das fases. Durante essa etapa foi utilizado o editor de mapas Tiled [14] para desenhar e exportar as configurações da fase para a Webty. Caso o motor já possuísse uma interface com editor de mapas, esse processo seria otimizado. Outro exemplo é a compressão dos arquivos finais do jogo, sendo necessário que o desenvolvedor realize a etapa de compressão por conta própria.

Apesar dos problemas já citados, pode-se afirmar que a Webty se saiu bem no seu primeiro teste, conseguindo produzir um jogo completo e com um certo sucesso comercial. Além disso, a capacidade do motor de adaptar o jogo criado a diferentes telas de forma automática se mostrou um sucesso.

O desenvolvimento do jogo Pimi Jumpers ajudou a afirmar o motor como um produto capaz de criar jogos comerciais, e apontou novos caminhos a serem seguidos.



Figura 5: Pimi Jumpers sendo executado nos dispositivos iOS, Android, Windows Phone, Firefox OS e PC

4 CONCLUSÃO

O motor Webty ainda está em desenvolvimento mas já se demonstra capaz de ser usado como uma ferramenta para qualquer um produzir jogos de forma comercial. Apesar do desenvolvimento de uma ferramenta desse porte ser algo desafiador, o processo tem se mostrado recompensador graças a quantidade de conhecimento adquirido.

Os resultados encontrados permitem traçar os próximos passos. Será dada continuidade as implementações de novos módulos e otimização dos já existentes. Também deverá ser investido tempo no estudo e desenvolvimento de uma interface gráfica para o

motor, visando torná-lo mais produtivo e ao mesmo tempo diminuir a curva de aprendizagem da ferramenta, aumentando assim o seu alcance.

REFERÊNCIAS

- [1] J. Fleming. Down the Hyper-Spatial Tube: Spacewar and the Birth of Digital Game Culture. 2007. Disponível em: <http://www.gamasutra.com/view/feature/129861/down_the_hyperspatial_tube_.php?page=2>. Acesso em: 27 jun. 2016.
- [2] J. Gregory. Game Engine Architecture. 2. ed. Florida: A K Peters/crc Press, 2014. 1052 p.
- [3] Michael Lewis and Jeffrey Jacobson. Game engines in scientific research. Communications Of The Acm, New York, v. 45, n. 1, p.27-31, Jan. 2002. Mensal.
- [4] T. O'Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. 2005. Disponível em: <<http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>>. Acesso em: 27 jun. 2016.
- [5] A. Toffler. The Third Wave. New York: Bantam Books, 1980. 560 p.
- [6] M. A. Rappa. The utility business model and the future of computing services. Ibm Systems Journal. Yorktown Heights, Ny, p. 32-42. jan. 2004.
- [7] A. Egges. Building JavaScript Games: for Phones, Tablets, and Desktop. New York, Ny: Apress Media, 2014. 444 p.
- [8] J. P. Zagal. et al. Towards an Ontological Language for Game Analysis. In: DIGRA INTERNATIONAL CONFERENCE: CHANGING VIEWS: WORLDS IN PLAY, 2., 2005, Vancouver, Canada. Proceedings... . Vancouver, Canada: Digra, 2005. v. 3.
- [9] T. Smith. The State of Responsive Design. In: MCANLIS, Colt et al. HTML 5 Game Development Insights. New York, Ny: Apress Media, 2014. Cap. 13. p. 211-220.
- [10] R. van Der Spuy. Advanced Game Design with HTML5 and JavaScript. New York, Ny: Apress Media, 2015. 552 p.
- [11] Yuriy Bura and Paul Coates. Pro Android Web Game Apps: Using HTML5, CSS3, and JavaScript. New York, Ny: Apress Media, 2012. 664 p.
- [12] H. Phillipe. Pimi Jumpers. 2015. Disponível em: <<http://www.ojogos.com.br/jogo/pimi-jumpers>>. Acesso em: 30 jun. 2016.
- [13] Hoowu Game. Hoowu HTML5 Game Contest Results. 2015. Disponível em: <<http://www.html5game devs.com/topic/14420-hoowu-html5-game-contest-results/>>. Acesso em: 2016 jun. 2016.
- [14] T. Lindeijer. Tile Map Editor. 2008. Disponível em: <<http://www.mapeditor.org/>>. Acesso em: 30 jun. 2016.