

Desenvolvimento de Jogo HTML5 Multiusuário para Rede Social Facebook

Trabalho de Conclusão do Curso de Tecnologia em Sistemas Para Internet

Guilherme Pera Renault
Orientador: Fabio Yoshimitsu Okuyama

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

guilhermep94@gmail.com,fabio.okuyama@poa.ifrs.edu.br

***Resumo.** Empresas independentes na área de jogos digitais dispensam grandes distribuidoras, e portanto precisam de outros modos para que seus produtos cheguem aos consumidores. Com o surgimento dos jogos em redes sociais, essas empresas ganharam uma excelente oportunidade de realizar a distribuição dos seus jogos de forma barata, pois através dessas comunidades é possível direcionar e selecionar o público alvo uma vez que se tem acesso aos seus perfis. Pretende-se descrever neste artigo o processo de desenvolvimento de um jogo HTML5 multiusuário, utilizando a API disponibilizada pelo Facebook tanto para realizar a autenticação dos usuários como aproveitar as informações relacionadas aos seus perfis da rede social. A aplicação será desenvolvida com as tecnologias HTML5, PHP, MySQL e Javascript, utilizando-se de solicitações assíncronas de informações.*

1. Introdução

O mercado de jogos digitais vem crescendo e se tornando uma área de grande importância para a economia global, tendo gerado 91,5 bilhões de dólares em receita ao longo do ano de 2015 [Newzoo, 2015]. Entretanto, uma determinada parcela desse capital provê de um pequeno número de organizações, o que dificulta a concorrência para empresas da área que não dispõem de tantos recursos para investimento.

Com isso, é muito comum que desenvolvedores independentes divulguem e disponibilizem seus produtos por meio das redes sociais, já que esse meio proporciona, com eficiência, um acesso simples, rápido e direcionado para o público alvo que tende a consumir esse produto. O Facebook por exemplo, possui um sistema de propagandas que possibilita que empresas criem anúncios, escolhendo o tipo de público a ser alcançado. Quando se acessa a rede social, esses anúncios são filtrados conforme o perfil do usuário e então exibidos.

Um exemplo do poder de divulgação através de redes sociais é o jogo Candy Crush Saga. Conforme reportagem da Folha de São Paulo (22/04/2013), o game lançado pela empresa King em 12 de abril de 2012 era inicialmente exclusivo para o Facebook e logo tornou-se um sucesso. Durante o ano de 2013 contou com mais de 50 milhões usuários mensalmente ativos, e passou a estar presente em diversas plataformas como IOS, Android, Windows Phone entre outros [LEONARDO, 2013].

Hoje, se um desenvolvedor independente quiser criar um jogo similar ao Candy Crush ele pode recorrer à ferramenta de divulgação do Facebook, informando que seu produto tem como público alvo pessoas com perfis similares aos dos usuários do famoso jogo da King.

2. Desenvolvimento de jogos com HTML5

Não existe um modelo ou padrão absoluto ao qual deve-se recorrer quando o assunto é o desenvolvimento de jogos com HTML5, mas é importante conhecer boas práticas a se seguir e tecnologias seguras para utilizar.

2.1 Materiais disponíveis

Diversos materiais estão disponíveis na internet para auxiliar desenvolvedores no processo de criação de jogos com HTML5, diferindo em pontos como tecnologias utilizadas e tópicos abordados, além de alguns apresentarem exemplos conexos para melhor demonstrar a integração das etapas.

O documento “HTML5 - O guia completo para a criação de jogos HTML5 com Canvas e SVG” [ROUSSET, 2016] pode ser encontrado na seção “Artigos técnicos para web” do site da MSDN. Neste guia o autor compartilha seus conhecimentos para descrever e exemplificar etapas essenciais no processo de desenvolvimento de jogos com HTML5. Os assuntos abordados variam desde a implementação dos elementos gráficos, utilizando as tecnologias SVG ou Canvas, até a criação de jogos multiusuários por meio de WebSockets.

Enquanto o material descrito anteriormente enfatiza o processo de desenvolvimento de um jogo, a documentação disponível no Facebook para desenvolvedores foca na questão da integração de aplicações com a rede social, explicando tópicos como: conexão de uma aplicação, utilização das funcionalidades expostas pelas APIs, pagamentos, entre outros.

2.2 Tecnologias para desenho gráfico com HTML5

As técnicas de desenho das imagens digitais podem ser divididas em duas categorias distintas: os bitmaps e os vetores. Tecnicamente esses dois formatos tem características muito diferentes, mas visualmente suas diferenças podem passar despercebidas. O bitmap representa uma imagem composta por pixels, e é utilizado normalmente quando deseja-se realizar desenhos com altos níveis de detalhes. Por serem formados pixel a pixel, as imagens bitmaps sofrem distorções consideráveis se tiverem seus tamanhos alterados [CROWTHER, LENNON, BLUE, WANISH, 2014].

Imagens ou gráficos vetoriais são formados por cálculos matemáticos executados pelo computador, e portanto perdem pouca qualidade quando tem seus tamanhos escalados. Entretanto, os vetores perdem em eficiência se comparados aos bitmaps quando os desenhos possuem altos níveis de detalhes.

Escolher quais tecnologias utilizar para realizar os desenhos gráficos de um jogo é um ponto crucial quando se trata da performance. No contexto do HTML5 existem duas tecnologias amplamente utilizadas para representações gráficas: o SVG e o Canvas.

“SVG (Scalable Vector Graphics) é uma tecnologia desenvolvida pelo W3C (World Wide Web Consortium) para criar gráficos vetoriais escaláveis, com base na linguagem XML. Define a sintaxe para criação de imagens estáticas capazes de serem estilizadas com CSS, de serem incorporadas à HTML e de receberem vários tipos de efeitos gráficos. Por ser uma aplicação XML a SVG proporciona o desenvolvimento de páginas web muito mais leves, rápidas e com alto potencial de interatividade. Ao contrário das imagens nos formatos GIF, JPEG e PNG as imagens SVG escalam apropriadamente em qualquer dispositivo sem perder resolução adaptando-se ao tamanho da área de visualização quer o dispositivo seja um telefone portátil, um monitor de grandes dimensões ou uma impressora de alta definição.” [SILVA, 2012].

“O Canvas é um elemento de marcação da linguagem HTML que tem como princípio fornecer desenhos e gráficos de alto desempenho interpretados via JavaScript, a serem exibidos em navegadores web.” [SOUZA, 2014].

Conforme mostrado na figura 1, ao ser comparado com o SVG o Canvas perde em questão de desempenho quando é utilizada uma maior área para a renderização gráfica, mas acaba ganhando quando muitos objetos são desenhados na tela. Isso ocorre pois o Canvas é apenas um container bitmap, enquanto o SVG precisa manter referências adicionais para cada objeto renderizado.

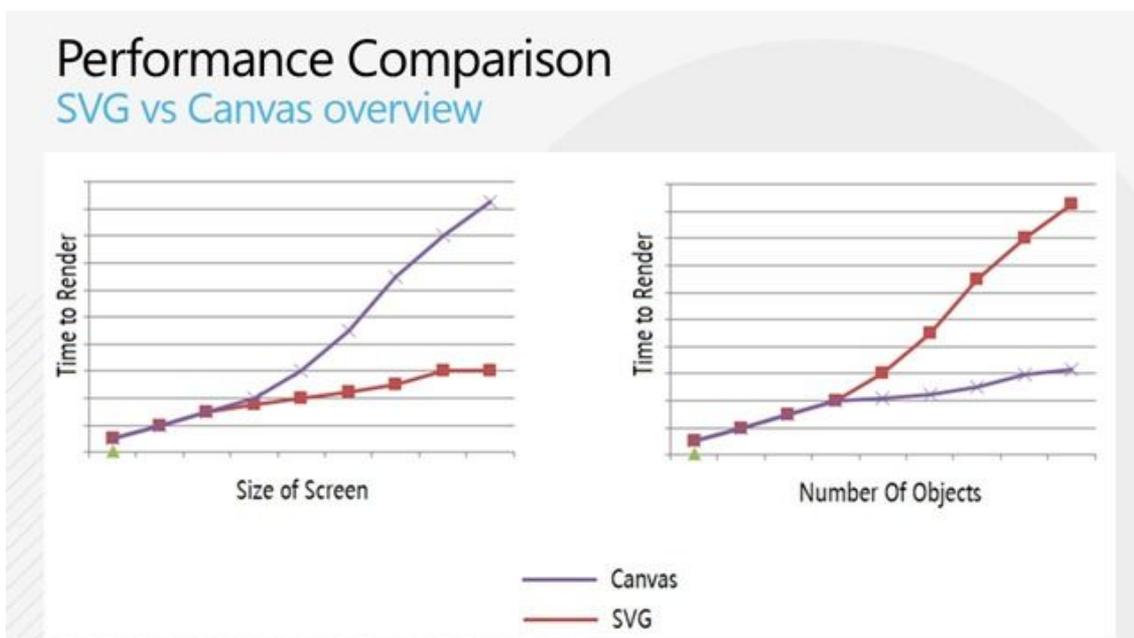


Figura 1. Comparação de performance entre SVG e Canvas [ROUSSET]

2.3 Comunicação cliente-servidor

A principal característica de um jogo multiusuário é a interação entre seus jogadores, que por sua vez nada mais é do que a troca de informações destes com o servidor. Alguns exemplos das possíveis interações são: conversas, listas de amigos e até mesmo pedidos de ajuda por meio de compartilhamentos em redes sociais (como no caso do jogo Candy Crush Saga).

Tratando-se de HTML5, as três formas mais conhecidas para prover essa comunicação cliente-servidor sem a necessidade de recarregar a página atual são o AJAX (Asynchronous JavaScript e XML), WebSockets e SSE (Server Sent Events). A figura 2 demonstra a diferença entre essas formas a respeito da comunicação cliente-servidor.

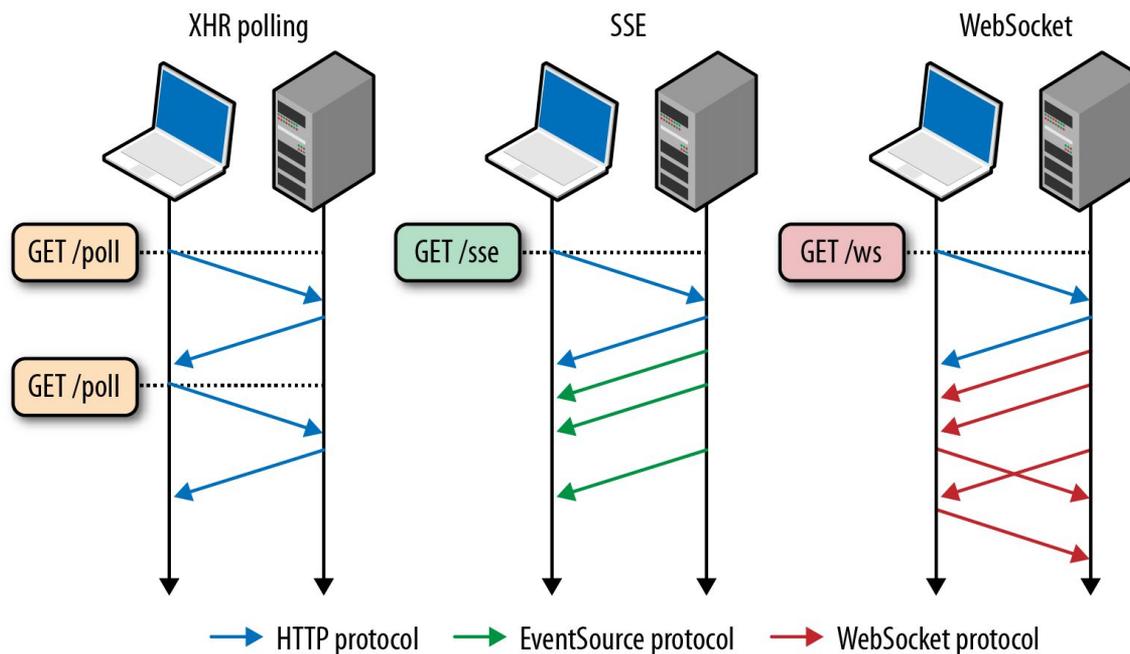


Figura 2. Communication flow of XHR, SSE, and WebSocket [GRIGORIK, 2013]

O AJAX dispõe do objeto XMLHttpRequest (XHR) do Javascript para se comunicar com scripts no lado do servidor. Esse objeto permite criar conexões, disparar requisições e receber seus devidos retornos, utilizando o protocolo HTTP. Com isso, é possível enviar requisições que ao serem acessadas satisfaçam necessidades específicas do cliente, como coletar ou atualizar as informações de um jogador no banco de dados. A principal característica do AJAX é a possibilidade de realizar essa comunicação cliente-servidor de forma assíncrona.

WebSockets são objetos expostos via Javascript que permitem estabelecer uma comunicação bidirecional e persistente entre o cliente e o servidor. Uma vez que se cria o canal de comunicação, ambas as partes podem começar a enviar dados a qualquer momento.

SSE, ou Server Sent Events, é um modo de conectar o cliente ao servidor por meio de uma conexão HTTP. Ele funciona de forma unidirecional, apenas recebendo informações do servidor, e por isso é mais eficaz quando se deseja atualizar constantemente as informações de uma página.

2.4. Autenticação com Facebook

Uma das etapas mais importantes quando se trata da integração de aplicações com a rede social é a autenticação de usuário, pois através dela é que se adquire acesso à

informações do perfil do mesmo. Para que esta autenticação seja possível, primeiro é necessário adicionar uma nova aplicação no Facebook Developers, escolhendo a plataforma (IOS, Android, Facebook Canvas ou Website), as informações que o usuário permitirá que sejam acessadas e outras diversas configurações. Após adicionada, o Facebook gerará um identificador para se referir à essa aplicação, possibilitando que as funcionalidades de integração disponíveis pela rede social sejam utilizadas.

Para que os desenvolvedores possam valer-se dessas funcionalidades, o Facebook Developers oferece diversos SDKs que facilitam a utilização de suas APIs em uma variada gama de plataformas. O código da figura 3, exposto na própria documentação do Facebook Developers, é um exemplo do carregamento e inicialização da SDK Web para Javascript.

```
<script>
  window.fbAsyncInit = function() {
    FB.init({
      appId      : 'your-app-id',
      xfbml      : true,
      version    : 'v2.6'
    });
  };

  (function(d, s, id){
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) {return;}
    js = d.createElement(s); js.id = id;
    js.src = "//connect.facebook.net/en_US/sdk.js";
    fjs.parentNode.insertBefore(js, fjs);
  })(document, 'script', 'facebook-jssdk');
</script>
```

Figura 3: Inicialização da SDK [Facebook Developers]

Uma vez carregado o SDK pode-se utilizar as funcionalidades das APIs. Outro exemplo que demonstra a simplicidade da integração da rede social com uma aplicação é a inclusão do plugin social que cria um botão de login, conforme a figura 4. Quando clicado, esse botão usa o SDK para apresentar uma interface e desencadear uma chamada para a API que trata da autenticação de usuários.

```
<fb:login-button scope="public_profile,email" onlogin="checkLoginState();" >
</fb:login-button >
```

Figura 4: Inserção de botão de login [Facebook Developers]

3. Desenvolvimento

Essa sessão tem como finalidade descrever a proposta do jogo que será utilizado como modelo, além das técnicas utilizadas e o porquê destas escolhas.

3.1 Descrição

O jogo a ser desenvolvido segue um modelo de progressão onde o jogador evolui seu personagem por meio de combates contra monstros. Caso o jogador vença o combate, ele é recompensado com uma quantia de pontos de experiência e moedas virtuais conforme o nível de dificuldade do combate. Conforme o personagem vai progredindo, é possível combater inimigos mais fortes que trarão recompensas melhores. A partir de um certo ponto o jogador pode desafiar outros jogadores com o propósito de pilhar suas moedas.

Uma interface possibilita que o usuário escolha o tipo de monstro e por quanto tempo irá caçá-lo, ou qual jogador irá combater. Quando confirmados pelo jogador, os combates são agendados para serem realizados de acordo com o tempo de caça escolhido pelo usuário, e podem ser cancelados caso o jogador mude de ideia. Um cronômetro mostra para o jogador quanto tempo falta até o próximo combate e, quando chegada a hora, a luta é executada no servidor de forma automática, sem a necessidade do usuário estar online.

Após executados, os resultados dos duelos podem ser visualizados em uma interface acessível aos jogadores. Cada monstro e personagem possui atributos próprios que definirão, juntamente com algumas variáveis aleatórias, o vencedor do combate. Esses atributos são melhorados conforme o personagem progride de nível ou gasta suas moedas para treiná-los. As moedas servem também para que o jogador aprimore seu portal de caça, liberando novos monstros para combater.

Existe também a possibilidade do jogador criar um grupo, convidando amigos diretamente do Facebook, o que incentiva a divulgação e compartilhamento do jogo. O propósito de um grupo é a cooperação entre os jogadores, que poderão investir em melhorias para todos integrantes. Além disso, fazer parte de um grupo possibilita que os participantes combatam juntos criaturas mais fortes do que o normal.

3.2 Casos de Uso

“Os casos de uso tornam claro o que o sistema irá fazer e, por omissão intencional, o que ele não vai fazer. Eles possibilitam a previsão efetiva, gerenciamento de escopo e desenvolvimento incremental de sistemas de qualquer tipo e tamanho” [JACOBSON, SPENCE e BITTNER, 2011]. O diagrama de casos de uso do sistema a ser trabalhado está representado na Figura 5.

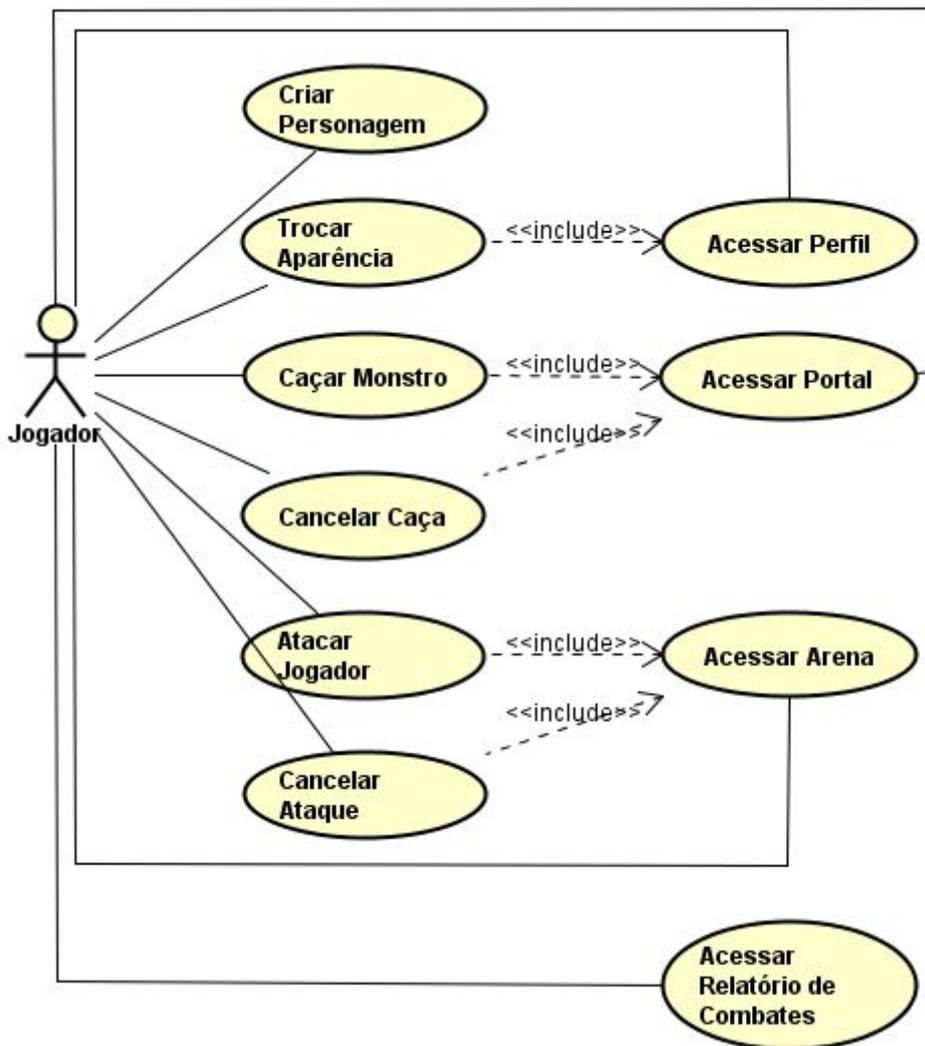


Figura 5. Diagrama de Casos de Uso

- Criar Personagem: O usuário cria seu personagem, selecionando entre as opções de classe e aparência.
- Acessar Perfil: O usuário acessa o perfil do personagem, podendo visualizar seus atributos.
- Acessar Portal: O usuário acessa a interface que permite agendar combates contra monstros.
- Acessar Arena: O usuário acessa a interface que permite agendar combates contra outros jogadores.
- Acessar Relatório de Combates: O usuário acessa a interface que permite visualizar os resultados dos últimos combates executados.
- Trocar Aparência: O usuário escolhe a aparência que representará seu personagem visualmente.
- Caçar Monstro: O usuário agenda um ou mais combates após escolher o tipo de monstro a ser caçado e quanto tempo irá caçar.

- Cancelar Caça: O usuário cancela todos os combates agendados que ainda não foram executados, retornando com o tesouro acumulado até então.
- Atacar Jogador: O usuário agenda um combate após escolher o jogador alvo.
- Cancelar Ataque: O usuário cancela o combate agendado contra um jogador.

3.3 Diagrama de Classes

O diagrama de classes representado pela figura 6 descreve a estrutura dos objetos da aplicação a serem persistidos.

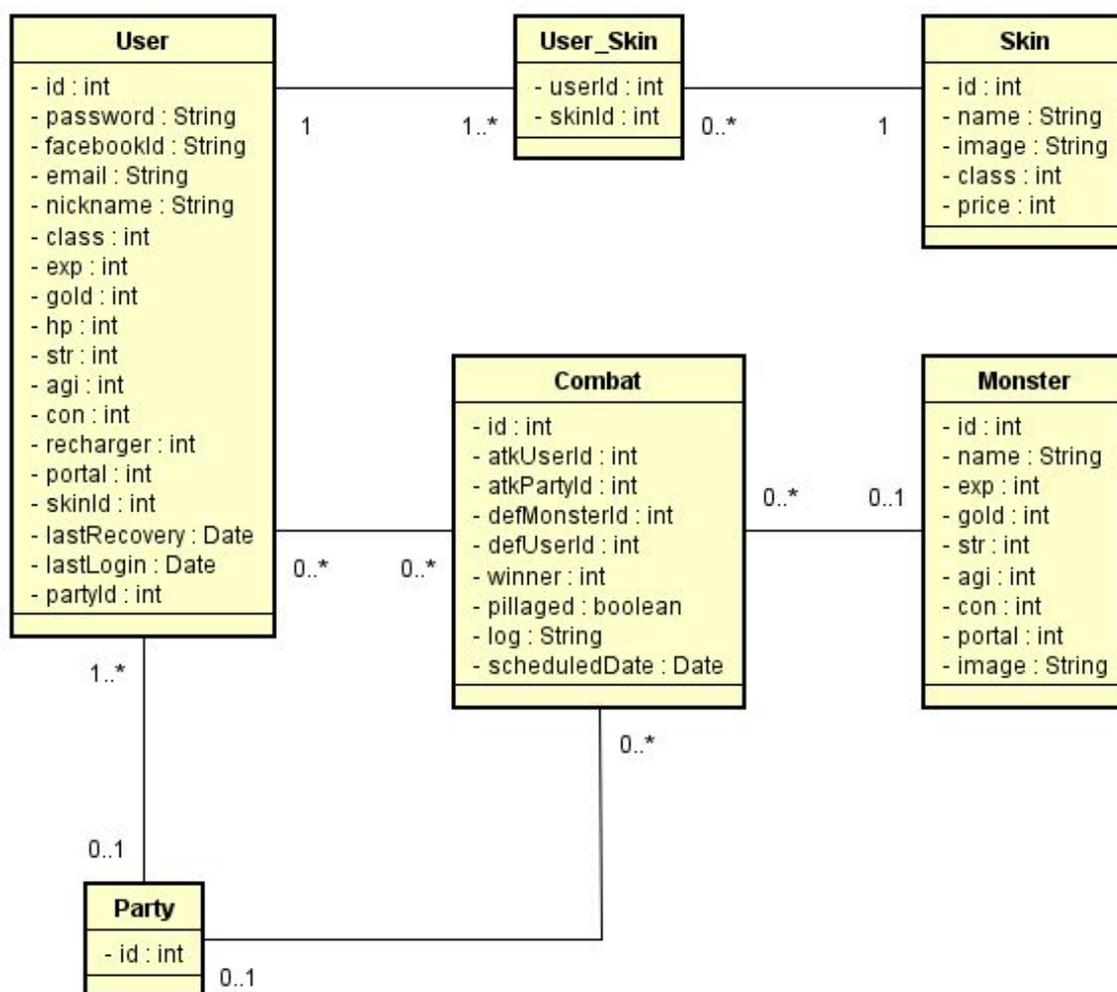


Figura 6. Diagrama de Classes

- User: Representa o usuário do sistema, armazenando os dados do seu personagem e cadastro.
- Skin: Diferentes visuais que um personagem pode utilizar.
- User_Skin: As skins que um jogador possui.
- Party: Grupo de jogadores que cooperam entre si.
- Combat: Representação do combate agendado por um jogador.
- Monster: Possíveis monstros para combater.

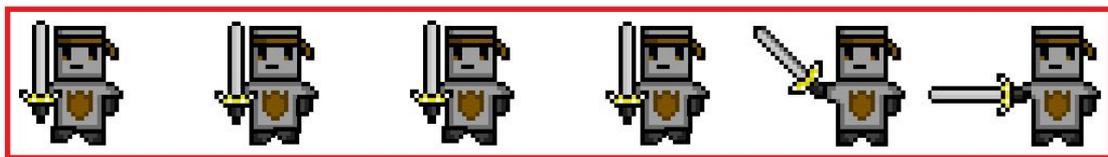
3.4 Consumo de Dados

O consumo de dados gerado por uma aplicação que requer conexão com a internet é um detalhe importante que muitas vezes acaba não recebendo a devida atenção, pois não é tão facilmente identificado e avaliado pelos usuários ou até mesmo pelo próprio desenvolvedor. Quando se trata de um dispositivo móvel, o racionamento do tráfego de dados é ainda mais importante. Isso ocorre pois boa parte da internet utilizada por estes dispositivos provêm de planos que cobram pelo volume de dados transferidos, podendo gerar custos adicionais para o usuário quando esta transferência excede os limites impostos pelo plano.

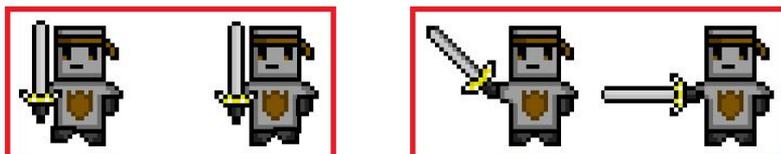
Pensando nisso, o sistema desenvolvido procura otimizar o consumo de dados do usuário através de técnicas e boas práticas. Um exemplo de uma técnica utilizada no projeto é o reaproveitamento de imagens, onde uma animação que repete trechos é fragmentada em sub animações. Esses trechos são então carregados pela aplicação uma única vez, e depois unidos novamente conforme a necessidade de repetições. Essa fragmentação funciona de forma similar à orientação a objetos, permitindo o reaproveitamento de estruturas.

A figura 7 demonstra o exemplo de fragmentação de uma animação em que o personagem acaba por repetir o movimento de caminhada. A animação completa é então dividida em duas partes, e a primeira é repetida apenas no momento da renderização gráfica. Nesse exemplo simples, a animação completa consome 6,5 megabytes e a fragmentada 4,8, gerando uma economia de aproximadamente 26% no consumo da internet.

Completa



Fragmentada



x2

Figura 7. Animação Fragmentada

Além de repetir a renderização de uma imagem na hora de montar uma animação, é possível utilizar essa mesma técnica para reutilizar os fragmentos em outras animações que possuem trechos em comum. No caso da figura 7, poderia ser reaproveitado o movimento de caminhada para criar uma animação na qual o personagem caminha e depois pula.

4. Considerações Finais

Levando em consideração o estudo necessário para a concretização deste artigo chegou-se a conclusão de que existem diversas formas de desenvolver um jogo para a plataforma web por meio do HTML5, e que o Facebook é uma ótima ferramenta para a divulgação desses jogos. Portanto, documentos que descrevam esse processo de desenvolvimento juntamente ao de integração da rede social são de grande importância para aqueles que buscam investir nessa área.

Sendo assim pretende-se que este material sirva de base para quem procura aprender sobre a área de desenvolvimento de jogos Web de forma barata e eficiente, a partir de uma ampla abordagem de tópicos e conhecimentos.

5. Referências

CROWTHER, Robe; LENNON, Joe; BLUE, Ash; WANISH, Greg. HTML5 em Ação. São Paulo: Novatec Editora, 2014.

Facebook Developers. *Quickstart: Facebook SDK for JavaScript*. Disponível em: <<https://developers.facebook.com/docs/javascript/quickstart>>. Acesso em junho de 2016.

Facebook Developers. *Facebook Login for the Web with the JavaScript SDK*. Disponível em: <<https://developers.facebook.com/docs/facebook-login/web>>. Acesso em junho de 2016.

GRIGORIK, Ilya. (2013) High Performance Browse Networking, Websocket.

JACOBSON, Ivar; SPENCE, Ian; BITTNER, Kurt. (2011) USE-CASE 2.0 - The Guide to Succeeding with Use Cases.

LEONARDO, Anderson. Game “Candy Crush” é sucesso no Facebook e em dispositivos móveis. Folha de S. Paulo, São Paulo, 22 de abril. 2013.

NEWZOO. (2015) Top 100 countries by 2015 game revenues. Disponível em: <<https://newzoo.com/insights/articles/newzoos-top-100-countries-by-2015-game-revenues/>>. Acesso em junho de 2016.

ROUSSET, David. HTML5 - O guia completo para a criação de jogos HTML5 com Canvas e SVG. Disponível em:

<<https://msdn.microsoft.com/pt-br/library/jj937876.aspx>>. Acesso em abril de 2016.

SILVA, Maurício Samy. Fundamentos da SVG. São Paulo: Novatec Editora, 2012.

SOUZA, Roque Fernando Marcos. Canvas HTML5. Rio de Janeiro: Brasport, 2014.

W3C. *Scalable Vector Graphics* (SVG). Disponível em:

<<https://www.w3.org/Graphics/SVG/>>. Acesso em maio de 2016.