

O navegador como plataforma para jogos: Uma experiência extracurricular para desenvolvimento de software

Augusto Henriques Marcus Vargas Tássio Auad Igor Knop

Centro de Ensino Superior de Juiz de Fora, Bacharelado em Sistemas de Informação, Brasil



Figura 1: Protótipo de jogo desenvolvido pelo mesmo grupo de alunos em dois momentos diferentes: desenho no elemento canvas através do uso de primitivas da API Javascript (à esquerda) em comparação com o uso de imagens (à direita).

Resumo

Este trabalho apresenta as atividades e resultados alcançados durante a criação de um grupo de estudos com o objetivo de utilizar a programação de jogos como ferramenta para a prática de desenvolvimento de software. Como plataforma utilizamos a API Javascript para desenho no elemento canvas, disponível no HTML5. Ao longo dos encontros os alunos conseguiram desenvolver um protótipo funcional de um jogo de ação e se mostraram muito interessados no formato das atividades.

Keywords: Desenvolvimento de Jogos, HTML5, Javascript

Contato dos autores:

{augustocesarinformatica, marcusvargasjf, auadtassio, igorknop}@gmail.com

1. Introdução

A criação de jogos apresenta um conjunto ímpar de desafios para projetistas e programadores, pois combina aspectos de narrativa, entretenimento, simulação computacional e interação com usuário [SARINHO e APOLINÁRIO 2008] para desencadear um conjunto de emoções no jogador, de forma a criar uma experiência divertida [SCHELL 2008]. Este complexo processo de desenvolvimento de software apresenta-se como um ótimo exercício para o treino de habilidades em programação em diversos estágios de conhecimento: há exemplos desde o uso de estruturas básicas de controle até técnicas avançadas de manutenção de código e busca constante por melhor desempenho de interfaces com o usuário. Estes tópicos são frequentemente abordados nos currículos tradicionais de Sistemas de Informação. Mas, por falta de experimentação, muitas vezes podem passar

despercebidos pelos alunos até a seu ingresso no mercado de trabalho.

A Internet é uma importante plataforma para divulgação e consumo de notícias, informação e entretenimento. Além do hipertexto e imagens, comuns em sua gênese, também são aceitos como recursos fundamentais para seu uso atual o vídeo, o áudio e as interações complexas com o usuário, na forma de aplicativos e jogos. Entretanto, os principais navegadores para acesso à rede não possuíam suporte nativo a estes elementos, necessitando instalação de programas de terceiros, na forma de plugins. Destes, podemos citar o Adobe Flash como o mais tradicional e popular para entrega de áudio, vídeo e para o desenvolvimento de jogos.

Por volta de 2004, a insatisfação com a direção dos trabalhos por parte do principal órgão para desenvolvimento da Web o World Wide Web Consortium [W3C 2011] levou à formação do grupo WHATWG [WHATWG 2011] que reunia os principais fabricantes de navegadores e grandes empresas ligadas à área de inovação baseadas na Internet como Apple, Opera, Mozilla e Google. O grupo possui como principal objetivo o contínuo desenvolvimento e evolução do padrão HTML, o que gerou uma nova versão do padrão, nomeada como HTML5 que atenderia às necessidades impostas pelas aplicações Web modernas. Entre os recursos mais comemorados pelos desenvolvedores de jogos está o suporte nativo por parte dos navegadores modernos a uma interface de programação de aplicativos (API, do inglês, Application Programming Interface) em Javascript que permite a criação de desenhos e animações de forma procedural.

Neste trabalho descrevemos as experiências e resultados obtidos com o uso do HTML5 como

plataforma para a criação de jogos de computador em uma atividade extra curricular para alunos do curso de Sistemas de Informação. Durante os encontros foi desenvolvido um protótipo funcional de um pequeno jogo de ação que pôde ser acessado via principais navegadores modernos, inclusive os disponíveis em dispositivos móveis.

2. O navegador como plataforma de jogos

Antes do advento do HTML5 as únicas opções para a distribuição de jogos via Internet estavam atreladas a manipulações da estrutura do HTML tradicional via Javascript ou pela utilização de plugins. Com o uso de um plugin, o navegador reserva um espaço na página onde um processo isolado consegue manipular imagens e realizar animações. O plugin Adobe Flash trabalha com desenhos vetoriais e animações construídas previamente ou que respondem a interações com o usuário programaticamente. Atualmente, o Flash é a plataforma mais popular para desenvolvimento de interfaces para jogos e animações para navegadores [BITTAR et al. 2010].

O HTML 5 traz consigo uma nova proposta para a construção de aplicações ricas para Internet, incluindo os jogos, onde o desenvolvimento pode ser feito utilizando padrões e tecnologias abertas, livres de licenças proprietárias. Nesta versão, o padrão HTML5 apresenta uma API Javascript para a construção de primitivas utilizando como área para desenho o elemento para canvas, embutido no código HTML da página. Assim é possível desenhar polígonos, elipses, imagens, textos e aplicar transformações nos eixos coordenados como translações e rotações. Com estas operações simples é possível criar jogos com os mais diversos graus de complexidade [DA SILVA e FIRMINO 2010].

3. Grupo de Desenvolvimento de Jogos

A fim de criar um espaço informal para os alunos praticarem o que aprendem nas disciplinas do curso, criamos um grupo de desenvolvimento de jogos objetivando, ao final dos encontros, o desenvolvimento de um protótipo funcional de um jogo.

3.1. Formato do grupo

Para início das atividades realizamos o convite para os alunos do primeiro período onde fixamos um encontro semanal, em um horário anterior ao início das aulas tradicionais. Optamos por começar as atividades após a primeira metade do semestre, quando os alunos já teriam sido apresentados às estruturas básicas de controle em algoritmos e o básico de Javascript. Entretanto, acabamos por receber alunos de períodos mais adiantados mas sem formação em Javascript devido à mudanças recentes na estrutura curricular.

Os encontros foram realizados semanalmente em intervalos de uma hora de duração, no laboratório de computação utilizando um único computador. Ao todo, foram oito encontros distribuídos em três meses. Durante os encontros, um par de alunos codificava e, em tempos regulares, revezavam com os demais para solucionar uma pequena funcionalidade definida no início do encontro. Este formato foi livremente baseado nos chamados Dojos de programação, mas sem o uso de técnicas de desenvolvimento orientado a testes.

Durante o desenvolvimento do protótipo, utilizamos o navegador Chromium como ambiente de desenvolvimento e o editor Gedit de um sistema Ubuntu GNU/Linux. A escolha do navegador foi motivada pelo suporte aos recursos mais novos do HTML5 e ao ótimo ambiente de acompanhamento de execução e depuração de Javascript. Utilizamos o sistema de controle de versão distribuído Git e todo o software desenvolvido durante os encontros pode ser obtido em <https://github.com/cesjf-gg/cesjfgg-2011-1> através da licença de código livre General Public License (GPL v3). Utilizamos imagens liberadas via Creative Commons obtidas em <http://opengameart.org>.

3.2. Tema para o protótipo

Apesar de fundamental para a construção de um jogo de qualidade, optamos por não definirmos um tema inicial nos primeiros encontros do grupo. Esperamos que o tema surgisse naturalmente após a experimentação dos recursos fundamentais de animação e controle de jogos por parte dos alunos.

Ao terceiro encontro, definimos um tema espacial para a construção do protótipo onde há um resgate de astronautas por uma nave espacial, com uma quantidade de combustível limitada e devendo evitar a colisão com asteroides. Dentro deste tema conseguimos explorar os fundamentos básicos de jogos como simulação de gravidade, controle de colisão, vidas, pontos, combustível, desenho dos elementos por primitivas e imagens, ciclo básico de animação, translação e rotação de eixos.

O protótipo consiste de uma nave que apenas pode ser acelerada verticalmente, consumindo combustível para vencer a força da gravidade que tenta jogá-la de volta à posição original. Dois tipos de elementos vagam em movimentos horizontais pela tela: astronautas e asteroides. Quando um astronauta é atingido pela nave este ganha uma certa quantidade de combustível e marca um ponto. Quando um asteroide é atingido gera-se uma animação de explosão, perde-se um ponto de vida. Então a nave retorna à posição inicial, abastecida.

3.3. Desenho de primitivas utilizando o contexto 2D

O trabalho iniciou com a construção de um documento HTML contendo um único elemento canvas para desenho de todos os objetos do jogo. A captura do objeto de contexto bidimensional em uma variável Javascript nos permite fazer o acesso à API de desenho de primitivas. A Listagem 1 ilustra um documento HTML básico utilizado para o desenho usando canvas.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>CESJF-GG: Protótipo 1</title>
  </head>
  <body>
    <canvas id="tela" width="300"
height="400">Sem suporte ao canvas. Atualize
seu navegador!</canvas>
    <script src="prototipo1.js" />
  </body>
</html>
```

Listagem 1

A Listagem 2 mostra um trecho em Javascript que captura o contexto bidimensional e realiza um desenho utilizando primitivas.

```
var tela = document.getElementById("tela");
var ctx = tela.getContext("2d");
ctx.beginPath();
ctx.rect(110+x,200+y, 40, 50);
ctx.closePath();
ctx.fillStyle = "rgb(150, 150, 250)";
ctx.fill();
ctx.lineWidth = 3;
ctx.stroke();
ctx.beginPath();
ctx.moveTo(110+x,250+y);
ctx.lineTo(90+x,250+y);
ctx.lineTo(110+x,235+y);
ctx.closePath();
ctx.fillStyle = "rgb(250, 150, 150)";
ctx.fill();
ctx.stroke();
```

Listagem 2

3.4 Laço principal de animação e captura de eventos

Para passar o efeito de movimento foi necessário criar um laço básico de animação onde os elementos do jogo teriam suas posições alteradas e depois desenhadas na tela. Durante a construção do protótipo utilizamos o método setInterval(), do objeto window, fornecido pelo navegador. Este método faz a chamada a uma função em um intervalo de tempo pré configurado como um temporizador. A Listagem 3 mostra o trecho de código onde fazemos o vínculo da função com o temporizador.

```
const fps = 42;
const segundo = 1000;
var intervalo = segundo/fps;
setInterval(passo,intervalo);
```

Listagem 3

Para a captura de eventos foram criadas duas funções: botaoPressionado(), que movimentará acelera a nave e botaoSolto() que cancela a aceleração forçada, deixando apenas a gravidade artificial atuar, levando-a até a posição inicial. A Listagem 4 apresenta o registro das funções com os eventos no navegador e um trecho com o teste para verificar qual tecla foi pressionada.

```
document.addEventListener("keydown",botaoPre
ssionado,false);
document.addEventListener("keyup",botaoSolto
,false);

function botaoPressionado(evento){
  if(evento.keyCode==38){
    acelerando=true;
  }
}
```

Listagem 4

3.5 Movimento de translação e rotação

Para o efeito de movimentação e de rotação de cada elemento do jogo, aproveitamos que o contexto 2D permite a translação e rotação dos eixos x e y antes de realizar o desenho. Para tanto, salvamos o estado dos eixos, realizamos a transformação de translação ou rotação, desenhamos o elemento e recuperamos o estado anterior dos eixos. A Listagem 5 mostra um trecho da função que desenha os astronautas.

```
function astronauta(x, y, a){
  ctx.save();
  ctx.translate(x, y);
  ctx.rotate(a*Math.PI/360);
  // (...) realiza o desenho
  ctx.restore();
}
```

Listagem 5

3.6 Colisão de objetos

O próximo passo foi criar as bases para a detecção de colisão de objetos móveis. Para gerar uma ação quando há colisão entre objetos, por exemplo, a colisão do objeto nave com um objeto astronauta ou asteroide, foi criada a função colisao() que recebe dois objetos contendo a posição, largura e altura. A Listagem 6 apresenta o código que realiza o teste de colisão retangular entre os objetos testando os casos onde não há colisões. Para todos os demais casos consideramos que há uma colisão entre os dois objetos testados.

```
function colisao(o1, o2){
  if((o1.y-o1.h/2)>(o2.y+o2.h/2)
  { return false;}
  if((o1.y+o1.h/2)<(o2.y-o2.h/2))
  { return false;}
  if((o1.x+o1.w/2)<(o2.x-o2.w/2))
  { return false;}
  if((o1.x-o1.w/2)>(o2.x+o2.w/2))
  { return false;}
```

```

return true;
}

```

Listagem 6

3.6. Animações de objetos

Quando há a colisão entre a nave e um obstáculo, disparamos uma explosão na posição atual. Utilizamos o desenho de um trecho da Figura 1 que contém vários quadros da animação de explosão dispostos em quatro linhas por quatro colunas.

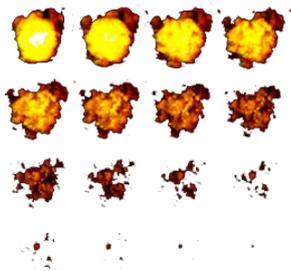


Figura 1: Quadros da animação de explosão.

O controle de qual quadro será desenhado é feito pelo trecho de código da Listagem 7, que apresenta como percorremos os dezesseis quadros da animação dispostos em grade.

```

function desenhaExplosao(x, y, a){
    ctx.save();
    ctx.translate(x, y);
    ctx.drawImage(imagemExplosao, frame*64,
frame1*64, 64, 64, -32, -32, 64, 64);
    ctx.restore();
}

```

Listagem 7

4. Considerações Finais e Trabalhos Futuros

Neste trabalho apresentamos as atividades e resultados obtidos com o uso de desenvolvimento de jogos como atividade extraclasse para motivar os alunos a exercitar o desenvolvimento de software, em especial a programação.

Utilizamos os novos recursos disponíveis no HTML5 para desenho e animações. Exploramos os navegadores modernos como plataforma de depuração e distribuição para implementação das funcionalidades básicas presentes na maioria dos jogos, sem utilização de quaisquer bibliotecas ou frameworks de terceiros.

Utilizar o navegador como ambiente de desenvolvimento e a API Javascript para manipulação do elemento canvas facilitaram as atividades pois não exigiram quaisquer configurações para uso imediato. Isto contribuiu para o entendimento por parte dos alunos iniciantes em programação e para alunos mais avançados que nunca tiveram contato formal com a linguagem Javascript.

Os encontros apresentaram uma dinâmica bem diferente da vista durante as aulas, onde os participantes do grupo buscavam soluções na base da tentativa e erro, se ajudando mutuamente. Ao final dos encontros relataram a experiência como recompensadora e conseguiram concluir um protótipo funcional, apesar do tempo restrito para as atividades.

Durante os encontros a troca de experiências entre alunos de períodos diferentes levantou várias questões importantes para desenvolvedores, como uso de ferramentas de controle de versão, quando utilizar recursos para refatoração e quando aumentar a abstração dos problemas.

Os alunos continuaram o desenvolvimento do protótipo enviando modificações, mesmo após o encerramento dos encontros, e propuseram várias ideias para continuação dos trabalhos. Dentre elas acrescentar progresso automático de dificuldade de acordo com a pontuação do jogador e integração com um servidor para registro de pontos.

Um ponto de interesse futuro, comum entre os membros participantes, é o desenvolvimento novos jogos explorando o contexto 3D com o WebGL [KHRONOS 2011].

Referências

- BITTAR, Thiago Jabur et al., 2010. *Considerações para jogos educativos na Web com base nas experiências e resultados do desenvolvimento do Ludo Educativo*. In: *Proceedings of SBGAMES2010 - IX Brazilian Symposium, SBC - Computing Track, Florianopolis, Brasil, November 8th-10th*. ISSN: 2179-2259.
- DA SILVA, Jucimar Maia; FIRMINO, Emiliano Carlos, 2010. *Desenvolvimento de Jogos em HTML5*. In: *Proceedings of SBGAMES2010 - IX Brazilian Symposium, SBC - Computing Track, Florianopolis, Brasil, November 8th-10th*. ISSN: 2179-2259.
- KHRONOS, 2011. *WebGL - OpenGL ES 2.0 for the Web*. Available from www.khronos.org/webgl [Accessed 7 august 2011].
- SARINHO, Victor T.; APOLINÁRIO, Antônio L., 2008. *A Feature Model Proposal for Computer Games Design*. In: *Proceedings of SBGames'08: Computing Track, Belo Horizonte, Brasil, November 10th-12th*. ISBN: 85-766-9204-X.
- SCHELL, Jesse, 2008. *The Art of Game Design, The book of Lenses*. Morgan Kaufmann. ISBN: 0123694965.
- WHATWG, 2011. *FAQ - WHATWG WIKI*. Available from: wiki.whatwg.org/wiki/FAQ [Accessed 5th august 2011].
- W3C, 2009. *About W3C*. Available from: www.w3.org/Consortium [Accessed 7 august 2011].