

Gráficos Vetoriais em Jogos

Achilles de Bylon Fróes Bisneto

Faculdades Jorge Amado
Av. Luís Viana Filho, n. 6775, Paralela, CEP: 41.745-130, Salvador, BA – Brasil
achilles.froes@gmail.com

Resumo

O presente artigo trata da utilização de gráficos vetoriais em jogos. Inicia-se apresentando um histórico do uso de gráficos. Em seguida, explica como classificar os tipos de games hoje existentes, bem como os possíveis tipos de gráficos em uso atualmente. Explana também o padrão *Scalable Vectorial Graphics* (SVG) no desenvolvimento de jogos e as vantagens de empregá-lo. Além disso, ilustra outras aplicações deste padrão. Por fim, apresenta o *toolkit batik* como a solução para a manipulação de imagens vetoriais em jogos.

Palavras-Chave: Jogos, gráficos vetoriais, toolkit, batik, SVG

Abstract

The present article deals about the use of vectorial graphics in games. It is initiated presenting a description of the use of graphics. After that, it explains how to classify the types of today existing games, as well as the possible types of graphics in use currently. Explains also the standard *Scalable Vectorial Graphics* (SVG) in the development of games and the advantages to use it. In additional, it illustrates other applications of this standard. Finally, it presents *toolkit batik* as the solution for the manipulation of vectorial images in games.

Keywords: games, vectorial graphics, toolkit, batik, SVG

1. Introdução

A visão talvez seja o sentido que mais nos norteia, nos preocupamos em ouvir o sons emitidos daquilo que estamos vendo, queremos tocar aquilo que vemos, sentir seu cheiro. Então não é de se admirar que demos tanta importância ao visual, a imagens! E imagens belas, por assim dizer, pois o que não nos agrada visualmente tentamos ignorar. Essa regra se aplica a vários pontos da vida humana, e não podia ser diferente no quesito entretenimento, mais especificamente jogos, jogos de computador.

Assim que começaram se difundir o que tornava um jogo atrativo era a simples diversão que ele proporcionava. Com o passar do tempo e o avanço da tecnologia, os jogos ganharam mais destaque e complexidade, e lógico, apelo visual.

O desenvolvimento de jogos deixou de ser responsabilidade apenas de

programadores para uma equipe com designers, escritores, sonoplastas, psicólogos. Entretanto, mesmo com todo esse detalhamento, no final das contas, um jogo é basicamente manipulação de gráficos mediante a regras bem definidas. E, para auxiliar essa criação e manipulação de gráficos, foi criado um conjunto de ferramentas, conhecidas no ramo como *toolkit*. Existem *toolkits* para diversas finalidades e talvez o mais importante seja o que lida com a parte gráfica do jogo.

Esses *toolkits* “podem ser encarados como ferramentas não convencionais que permitem ao artista transcender das técnicas tradicionais de desenho ou modelagem. Imagens que exigiriam do artista o uso de uma técnica apurada de desenho podem ser geradas mais facilmente com auxílio de softwares.” [...] “Com o uso de software, o artista precisa apenas a idéia e deixar a parte mais complexa por conta da máquina. Contudo, esses softwares exigem um certo nível de conhecimento e treinamento que forçarão os artistas a uma complementação do estudos de técnicas de desenho tradicional, com a teoria de computação gráfica e matemática” [8].

2. Gráficos em jogos, uma evolução através do tempo.

O desenvolvimento de gráficos para jogos começou com a intenção do físico Willy Higinbotham de atrair visitantes ao Brookhaven National Laboratories, no estado de Nova Iorque. Para tal feito ele criou um jogo de tênis: o Tennis Programming, que na época (1958 – Guerra Fria), era exibido em um osciloscópio e processado por um computador analógico. Foi um sucesso, e em pouco tempo o público não estava nem um pouco interessado em saber sobre megatons e fissura de átomos.

Em 1962 novamente com o intuito de atrair público, dessa vez o MIT (Massachusetts Institute of Technology), para conhecer o DEC PDCP-1, o primeiro microcomputador fabricado, que custou US\$ 120 mil dólares e tinha absurdos 4k de memória (muito para a época), cartões perfurados, monitor, um poderoso processador de 16 bits e uma caneta ótica. Desenvolvido em puro Assembly por Stephen Russell, inspirado num livro de ficção científica, - o Spacewar! - é considerado o primeiro jogo da história, ao invés do Tennis Programming.



Figura 1. Spacewar!, considerado o 1º jogo eletrônico.

Passados quatro anos, em 1968, o funcionário da Sander Associates, Ralph Baer, teve a brilhante idéia de criar uma máquina barata capaz de executar jogos eletrônicos utilizando uma televisão como saída. Nasce então o video-game "Brown Box" (seria o uma versão jurássica do atual "X-Box"?), com seu primeiro jogo, um rudimentar "Ping Pong". Além deste, houve jogos de futebol, voleibol e até mesmo de tiro. Contudo o primeiro vídeo game a ser lançado no mercado foi o Odyssey 100, fabricado pelo subsidiária Magnavox, da Philips holandesa, em 1972.



Figura 2. Pong, um jogo clássico, talvez o responsável disseminação dos jogos eletrônicos.

Novembro de 1977: início da era de ouro dos vídeo games. A Warner Communications, que acabara de comprar a Atari, lança o Atari VCS (Video Computer System), que mais tarde mudou seu nome para Atari 2600, com seus 128 bytes de memória, 1.19 Mhz de velocidade do processador e da placa de vídeo.

Por serem relativamente fáceis de criar e manipular (leia-se baixo processamento para animar), os gráficos bitmap foram amplamente utilizados principalmente pelo Atari 2600, e seus concorrentes. Até que em 1982 foi lançado o Vectrex, projetado para mostrar imagens vetoriais, ao invés de pixels, como era a maioria dos vídeo games, utilizando um processador da Motorola, o 68A09 (1,5 MHz). Felizmente, por questões de custo, ele só exibia imagens em preto e branco. Até mesmo antes do Vectrex, jogos com gráficos vetoriais foram lançados, a exemplo do Victronics' *Warrior* (1979): um jogo de luta, e o *Tempest* (1980), um jogo de nave para o Atari 2600. Contudo o Vectrex foi o primeiro projetado especialmente para gráficos vetoriais. Infelizmente não teve sucesso comercial por ser caro e ter seus gráficos monocromáticos.

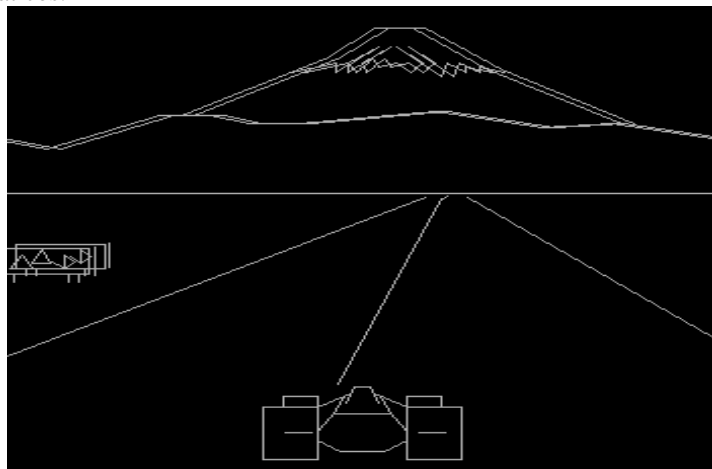


Figura 3. Jogo de corrida desenvolvido para o Vectrex.

Devido à superpopulação de consoles e jogos sem qualquer inovação, houve, em 1984, o que se chamou de "crash dos videogames". Ainda assim, o MSX (1983) equipado com o excelente processador Z-80, da Zilog, placa de vídeo Texas Instruments com 16 KB de memória, 32 KB de BIOS (Basic), 2 portas de expansão, 1 saída para joystick/mouse e conexão para gravador e o MSX 2.0 com 64 KB, no mínimo de RAM, VRAM (pelo menos 64 KB), um relógio interno em tempo real, uma Sub-Rom de 16 KB (além da Bios de 32 KB) e a placa de vídeo da Yamaha, a nova V9938, capaz de mostrar até 256 cores simultâneas merecem destaque, por serem vídeo games em corpo de computador e possuem jogos incríveis, como Metal Gear e Antarctic Adventure.

Em 1985, o mercado se recupera: nasce a gigante Nintendo Entertainment System, responsável vídeo game NES, que dominou 90% do mercado da era 8 bits até o início dos anos 90, com jogos criativos e cativantes como: The Legend of Zelda, Megaman e Metroid. Todos eles utilizavam bitmaps.

Lançado pela Sega em outubro de 1988, o Mega Drive foi o primeiro vídeo game de 16bits, equipado com o Motorola 68000, muito poderoso, que rodava a grandiosos 7.67 Mhz. Graças a ele, os gráficos vetoriais foram amplamente utilizados em jogos, como: Space Harrier 3D, Virtua Fighter, Vector Man e Virtua Racing, que foi predecessor de jogos poligonais.



Figura 4. Vitua Racing, o precursor da nova geração no uso de gráficos poligonais utilizando vetores.

3. Tipo de jogos

Existem algumas formas de classificar um jogo:

- Quanto ao gênero ou categoria: ação, aventura, corrida, erótico, esporte, estratégia, exótico, luta, MMO (*Massive Multiplay Online*), MMOFPS (*Massive Multiplay Online First Person Shooter*), MMORPG (*Massive Multiplay Online Role Play Game*), musical, *Party Game*, *Pinball*, plataforma, *puzzle*, RPG (*Role Play Game*), simulação, survival horror, tiro, tiro em primeira pessoa.
- Quanto tecnologia usada: envolve basicamente a biblioteca gráfica utilizada, podem ser: opengl, directx, sdl, allegro, lwjgl, jol, flash.
- Quanto ao ambiente: desktop, web e mobile.

Cada classificação possui suas peculiaridades e podem ser combinados, criando, desta forma, uma gama de jogos para todos os gostos. Por exemplo: pode existir um jogo de corrida para desktop desenvolvido em opengl. No mercado de jogos existem combinações altamente saturadas, a exemplo dos jogos de tiro em primeira pessoa e estratégia para desktop desenvolvidos em directx.

Atualmente muitas *softhouses* (como são conhecidas as empresas que desenvolvem jogos) tentam inovar criando jogos criativos e simples que atraiam a atenção dos jogadores casuais. Estes são os que jogam menos de 1h por dia e de forma esporádica. Esse perfil de jogador procura por jogos simples, que proporcionem diversão rápida e de forma descomplicada.

A maioria dos jogos que esse tipo de jogador procura são os diversos gêneros para web e desenvolvidos em flash. A vantagem dos *web games* (como são chamados os jogos para web), é que apenas precisa instalar o *plug-in*. Estes jogos são acessados de forma simples pelo navegador e assim que o carregamento termina, instruções simples são passadas e a diversão começa.

4. Tipos de gráficos

Um jogo eletrônico nada mais é do que a exibição de gráficos pré-renderizados, ou renderizados em tempo real. Existem dois tipos de gráficos.

O gráfico tipo *Raster* é pré-renderizado e composto por pixels. São comumente chamadas de bitmap [6]. Por sua vez, o bitmap consiste em uma matriz de pontos armazenado na memória. Cada ponto da matriz é chamado de pixel, seguindo o padrão RGB ([red](#), [green](#) e [blue](#)). Cada pixel pode definir 256 cores utilizando 8bits para armazenar a informação da cor. Aumentando a resolução (quantidade de pixels utilizados) e a quantidade de bits para informar a cor, facilmente ultrapassa a quantidade de cores que olho humano é capaz de perceber. Mas isso tem um custo: quanto maior a qualidade da imagem, maior será o espaço ocupado na memória e mais trabalhoso será manipular uma imagem desse porte. Bitmaps são muito utilizados para fotos digitais e nos jogos normalmente representam as texturas dos objetos, o que seria muito complexo fazer com imagens vetoriais. Os principais formatos são: JPEG, TIFF, PNG, GIF, BMP, XPM.

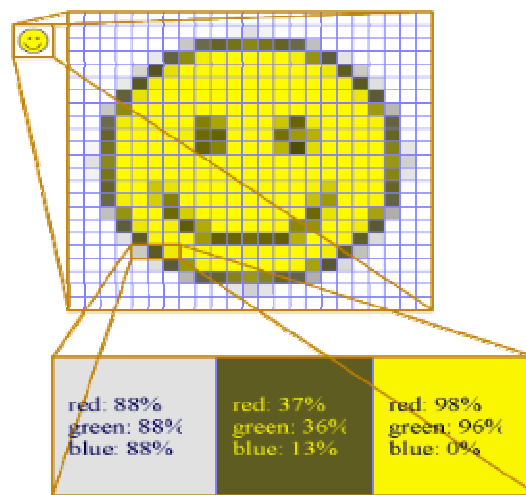


Figura 5. Exemplo de imagem bitmap.

Já o gráfico tipo vetorial é renderizado em tempo real, e composto por descrições geométricas). Uma imagem vetorial é um tipo de imagem gerada a partir de descrições geométricas de formas, diferente das imagens chamadas mapa de *bits*, que são geradas a partir de pontos minúsculos diferenciados por suas cores. Uma imagem vetorial normalmente é composta por curvas, elipses, polígonos, texto, entre outros elementos, que utilizam vetores matemáticos para sua descrição. Em um trecho de desenho sólido, de uma cor apenas, um programa vetorial apenas repete o padrão, não tendo que armazenar dados para cada pixel.

Por serem baseados em vetores, esses gráficos geralmente são mais leves (ocupam menos memória no disco) e não perdem qualidade ao serem ampliados, já que as funções matemáticas adequam-se facilmente à escala, o que não ocorre com gráficos *bitmap*, por utilizarem métodos de interpolação na tentativa de preservar a qualidade.

Outra vantagem da imagem vetorial é a possibilidade de isolar objetos e zonas. Cada objeto é uma entidade independente com propriedades como cor, forma, traçado, tamanho e posição na tela, incluídas na sua definição.

Visto que cada objeto é uma entidade independente, é possível mover e alterar suas propriedades repetidas vezes e manter a sua nitidez e resolução originais, sem afetar os demais objetos na ilustração. Os formatos mais utilizados são: ps (*PostScript*), eps (*Encapsulated PostScript*), ai (*Adobe Illustrator Files*), fig (*Xfig files*), wmf (*Window's MetaFile*- somente parcialmente vetor), svg (*Scalable Vector Graphic*).



Figura 6. Exemplo de imagem vetorial.

5. O padrão SVG

SVG é a abreviatura de *Scalable Vectorial Graphics* que pode ser traduzido do inglês como gráficos vetoriais escaláveis. Trata-se de uma linguagem XML para descrever de forma vetorial desenhos e gráficos bidimensionais, quer de forma estática, dinâmica ou animada. A grande diferença entre o SVG e outros formatos vetoriais, é o fato de ser um formato aberto, não sendo propriedade de nenhuma empresa. Ela foi criada pela World Wide Web Consortium, responsável pela definição de outros padrões, como o HTML e o XHTML.

SVG permite três tipos de objetos gráficos:

- Formas vetoriais (ex: linhas, curvas)
- Imagens *raster* (bitmap)
- Texto

Objetos gráficos podem ser agrupados, estilizados, transformados e compostos em objetos previamente rederizados. O texto pode ser qualquer *namespace* XML apropriado à aplicação, que permita pesquisa e acesso aos gráficos SVG. O conjunto de características incluem transformações aninhadas, *clipping paths*, *alpha masks*,

filtros de efeito, objetos de *template* e extensibilidade.

Os desenhos SVG podem ser dinâmicos e interativos. O *Document Object Model* (DOM) para o SVG inclui o XML DOM completo, permitindo animação direta e eficientemente dos gráficos vetoriais via ECMAScript, SMIL ou qualquer outra linguagem interpretada. Um rico conjunto “capturadores” de eventos como *onmouseover*, é o evento de quando o ponteiro do mouse passa por cima do objeto, e *onclick*, como o próprio nome sugere captura o evento de clique do mouse no objeto, podem ser designados a qualquer objeto SVG. Por causa dessa compatibilidade e suporte de outros padrões *web*, recursos como scripting, programar eventos e animações, pode ser feitos nos elementos SVG e outros elementos XML de diferentes *namespaces* simultaneamente no mesmo arquivo.

Uma vantagem do padrão é em relação ao tamanho dos arquivos, as imagens SVG podem ser salvas de forma compacta usando *gzip*, programa para compactação de arquivos. Nesse caso eles são chamados de “arquivos SVGZ”. Por ser composto basicamente de texto, os arquivos XML tendem a reduzir bastante e esses arquivos ficam muito pequenos. De qualquer forma, os arquivos SVG já são menores que a versão *raster*, que seria um mapa de bits armazenando as informações da imagem desejada.

5.1 Aplicações do SVG

Atualmente o SVG é utilizado em dispositivos móveis, impressões em grandes resoluções, aplicações web, design e interoperacionalidade, GIS e *Mapping*, sistemas embarcados.

Em 2001, a indústria de telefonia móvel escolheu SVG como base para plataforma gráfica. As companhias principais se juntaram para desenvolver os perfis SVG Tiny e o SVG Basic, comumente chamado de SVG Mobile, objetivando dispositivos com recursos limitados como *handsets* e PDAs.

A combinação de características gráficas ricas, suporte a texto compreensivo e a independência de resolução um formato ideal para impressão. As principais companhias de impressoras atualmente estão desenvolvendo a especificação SVG Print: uma versão especialmente desenvolvida para impressão profissional.

Aplicações web estão em crescente popularidade: os desenvolvedores são frequentemente limitados pelas incompatibilidades dos navegadores - como Internet Explorer, Firefox, Opera - e falta de funcionalidades. Com suporte a scripintg e captura de eventos, SVG pode ser usado como uma plataforma para desenvolvimento de aplicações gráficas e interfaces com o usuário.

Por ser estendido do XML, os diagramas SVG permitem que seja definido um metadata, embarcado em formatos proprietários sem comprometer a apresentação. Por exemplo: um programa CAD poderia exportar para SVG uma exibição online, mas armazena dados no arquivo que facilita no futuro a edição ou roundtripping, conversão de um formato de arquivo para outro (como pdf para html). Sendo assim, desde que as ferramentas dêem suporte a importação e exportação em SVG, ele pode ser usado como meio de interoperacionalizar aplicações em um formato em comum.

O GIS - *Geographic Information Systems* – tem requisitos muito específicos: recursos gráficos ricos, suporte a conteúdo em vetor e bitmap e habilidade para armazenar uma grande quantidade de dados. SVG é compatível a esse tipo de

finalidade e muitos sistemas GIS provêem exportação em SVG.

6. O Batik SVG Toolkit – uma ferramenta para manipular arquivos SVG

O Batik é um toolkit baseado na tecnologia Java para aplicações que pretendem usar o SVG para vários propósitos, como visualização, geração e manipulação.

Esse toolkit provê ao desenvolvedor uma coleção de módulos-núcleo que podem ser usados em conjunto ou individualmente, para suportar soluções específicas em SVG.

Com o Batik é possível manipular documentos SVG em qualquer lugar onde a tecnologia Java esteja disponível. Ele torna fácil para aplicações Java utilizar o conteúdo SVG.

A última revisão do Batik está conforme a especificação estática do SVG e suporta os recursos de interatividade, linking e scripting da especificação SVG. O projeto Batik não provê suporte a animação declarativa e não há atualmente planos adicionar esse tipo de suporte.

7. Conclusão

Apesar do padrão SVG ainda não ser utilizado no desenvolvimento de jogos, é uma alternativa viável, pois possui vantagens que outros formatos de imagem vetorial não possuem. Além disso, possui vasta documentação, por ser um padrão descrito por um consórcio (W3 Consortium) que já especificou outros padrões da Web, como HTML e XML. Neste último caso, é do onde o SVG se deriva. Desta forma, juntamente com a utilização do batik, o SVG torna-se viável para o desenvolvimento de jogos, pois facilita este processo.

8. Referências

[1] (SVG, 2006) World Wide Web Consortium. “Scalable Vector Graphics (SVG) - XML Graphics for the Web”. Disponível em <<http://www.w3.org/Graphics/SVG/>> Acessado em 25 de setembro de 2006.

[2] (Learn SVG, 2006) http://www.w3schools.com/svg/svg_intro.asp

[3] (Vector graphics, 2006) Wikipedia, the free encyclopedia. “Vector graphics”. Disponível em <http://en.wikipedia.org/wiki/Vector_graphics> Acessado em 20 de outubro de 2006.

[4] (Outerspace, 2006) A História dos Videogames. Disponível em

<<http://outerspace.ig.com.br/retrospace/>> Acessado em 23 de outubro de 2006.

[5] (UNIFESP Virtual, 2006) Imagens digitais. Disponível em <<http://www.virtual.epm.br/material/ead/imagem.htm>> Acessado em 25 de outubro de 2006.

[6] (Mike Sketchpad, 2006) Two Kinds of Computer Graphics. Disponível em <<http://www.sketchpad.net/basics1.htm>> Acessado em 25 de outubro de 2006.

[7] (Webopedia, 2006) Bitmap. Disponível em <http://www.webopedia.com/TERM/b/bit_map.html> Acessado em 25 de outubro de 2006.

[8] [Azevedo, Eduardo. Computação Gráfica – Teoria e Prática. Rio de Janeiro: Elsevier, 2003.](#)