

# Construção de um sistema de visualização de arquivos 3D

**Mauro Marcelo Mattos, Antonio Carlos Tavares, Vinícius Campos da Rosa  
Krauspenhar**

Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil  
`{mattos, tavares}@inf.furb.br, vinicius.krauspenhar@benner.com.br`

**Abstract.** *This paper describes a project that aims to build a 3D visualization of a Windows file system. The main concepts are presented and the system architecture is described.*

**Resumo.** *O presente trabalho apresenta a construção de um sistema de visualização de arquivos em um sistema de arquivos Windows utilizando uma metáfora 3D. A arquitetura do sistema é descrita bem como os resultados alcançados.*

## 1. Introdução

As pesquisas em visualização de um sistema de arquivos utilizando metáforas 3D não são recentes. Vários aspectos são envolvidos num projeto neste contexto. Dentre os principais pode-se citar: notificação e sincronização entre objetos do sistema de arquivos e objetos da camada de visualização; sincronização entre ações da camada de visualização e a efetivação na camada do sistema de arquivos; impacto de performance no sistema e escolha da metáfora de representação gráfica. Estas questões são abordadas no presente projeto.

Gerenciar o armazenamento e a recuperação de documentos de forma eficiente nesse ambiente tem sido um problema constante para a maioria dos usuários, agravada pela grande quantidade de arquivos disponíveis. Farhoomand e Drury (2002) concluem em sua pesquisa que 62% dos usuários têm dificuldade ou impossibilidade de gerenciar seus documentos.

Robertson et al.(1998) descrevem o projeto Data Mountain o qual consiste em uma interface criada especialmente para aproveitar a habilidade humana de guardar informações sobre um determinado ambiente e sua orientação espacial. A neurociência trata esta habilidade como memória espacial (UNIVERSITY OF BRISTOL, 2001).

Analizando o problema apresentado na gerência de arquivos e os resultados apresentados pelo projeto descrito, surge a idéia de criar um ambiente de visualização 3D de um sistema de arquivos baseado nos princípios de memória espacial. A ferramenta é voltada para o sistema operacional Windows e pode ser utilizada nos sistemas de arquivos Fat32 e NTFS. A ferramenta funciona em placas de vídeo com 64 MB de RAM que suportem OpenGL ou DirectX 9 ou superior. No contexto deste projeto, o conceito da metáfora está relacionado à utilização de imagens como forma de permitir ao usuário manipular componentes do sistema de arquivos.

O trabalho está organizado da seguinte forma: a seção 2 apresenta alguns

conceitos de visualização de informações. A seção 3 apresenta alguns trabalhos correlatos. A seção 4 descreve a arquitetura do sistema e seção 5 finaliza o artigo apresentando as principais conclusões e indica alguns trabalhos futuros.

## 2. Visualização de informações

A área de visualização de informações utiliza sistemas gráficos 3D e técnicas de animação interativa para simular o reconhecimento de padrões e estruturas em informações armazenadas (Robertson et al., 1998). Conforme Freitas et al. (2001, p. 143), “visualização de informações é uma área de aplicação de técnicas de computação gráfica, geralmente interativas, visando auxiliar o processo de análise e compreensão de um conjunto de dados, através de representações gráficas manipuláveis”.

Dentre as principais técnicas de visualização podem-se citar: Apperley, Tzavaras e Spence (1982) apresentaram a técnica Bifocal Display (figura 1a). Mackinlay et al. (1991, p. 173) apresentaram a técnica Perspective Wall (figura 1b) e Bederson (2000) descreve um esforço no sentido de aplicar técnicas tradicionais de visualização gráfica baseadas em distorção, do tipo fisheye (figura 1c), para tornar mais eficiente o mecanismo de seleção de itens de menus quando a lista de opções é muito grande.

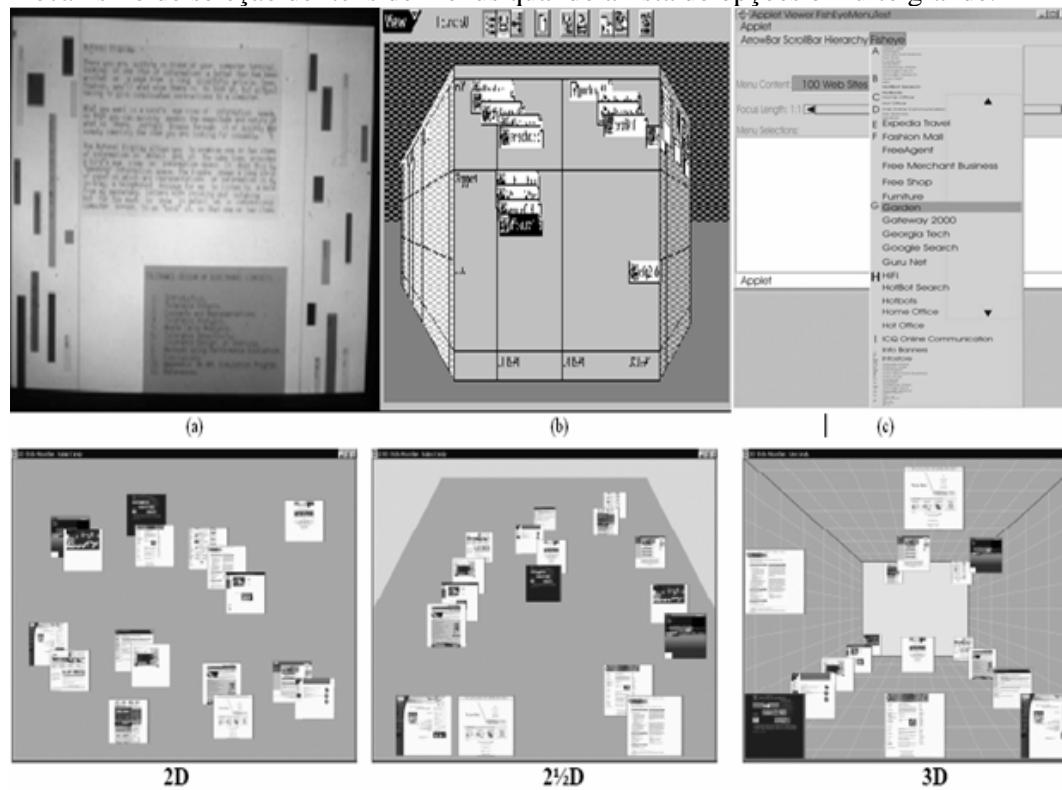


Figura 1 – (a) Representação visual da técnica Bifocal Display. (b) Representação visual da técnica Perspective Wall no software Visual Recall (c) Fisheye menu relacionando os 100 mais populares sites web da lista dos mais populares da revista PC Magazine.

### 2.1. Memória espacial em ambientes virtuais

Cockburn e McKenzie (2001, p. 138) chegaram a algumas constatações ao analisar o modelo proposto pelo projeto Data Mountain (figura 1). Utilizando um ambiente virtual

vários usuários foram submetidos às tarefas de organizar páginas e posteriormente acessar as mesmas na organização criada. A maioria dos usuários apresentou dificuldade na tarefa de organizar, porém, ficou comprovada a capacidade espacial do cérebro humano de acessar rapidamente os documentos. A utilização de um ambiente tridimensional também apresenta algumas facilidades, como a possibilidade de movimento e rotação, permitindo alterar o ponto de vista das informações (COCKBURN; MCKENZIE, 2001, p. 145).

### 3. Trabalhos correlatos

Robertson et al. (1999) descrevem o desenvolvimento de um projeto da Microsoft para construir uma interface para expandir a janela do desktop em um ambiente 3D com ilimitadas janelas de desktop denominado Task Gallery (ROBERTSON et al., 1999). A proposta foi criar uma ilusão baseada na habilidade humana de visão espacial, aonde as pessoas vão intuitivamente navegar no ambiente. Os estudos utilizando o Task Gallery demonstraram que os usuários utilizaram o sistema de forma bastante natural, tendo comparado o ambiente ao mundo real (ROBERTSON et al., 1999, p. 5). Frutos dessa pesquisa podem ser encontrados no Windows Vista, como por exemplo o gerenciador de janelas TopDesk.

A interface apresentada pelo Task Gallery utiliza a técnica Perspective Wall apresentada por Mackinlay et al. (1991). O projeto BumpTop (AGARAWALA; BALAKRISHNAN, 2006, p. 1) consiste em um ambiente de gerência de arquivos em 3D para o sistema operacional Windows utilizando conceitos de física na manipulação de documentos, como atrito e massa. Utiliza também um conceito de pilhas para organizar documentos.

Diversas ferramentas foram desenvolvidas, entre elas, pode-se citar: (i) Tactile 3D (UPPER BOUNDS INTERACTIVE, 2006) onde é apresentado um ambiente semelhante a um jogo tridimensional; (ii) a ferramenta XCruiser (SHINYAMA, 2003) aplica a metáfora de galáxias para o sistema de arquivos, onde em um ambiente 3D, arquivos são representados como planetas e diretórios por galáxias.

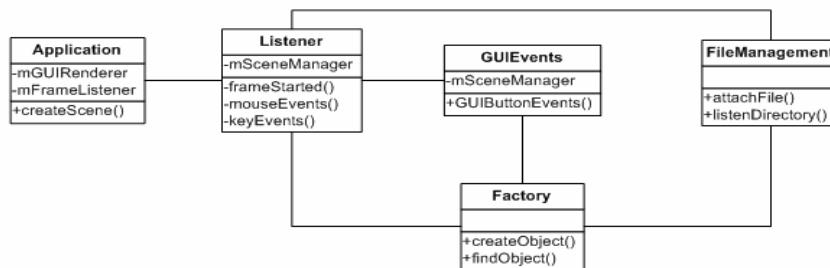
### 4. Arquitetura do Sistema

O sistema está sendo desenvolvido com o intuito de prover um ambiente tridimensional para representar a manipulação de um sistema de arquivos. Para prover a interface gráfica, está se utilizando a OGRE (Object-Oriented Graphics Rendering Engine). Este motor gráfico auxilia o desenvolvimento de aplicações que utilizam gráficos 3D. A biblioteca de classes da OGRE abstrai todas as funções disponibilizadas por bibliotecas mais complexas como OpenGL e Direct3D.

Para viabilizar a manipulação desta interface através de dispositivos de entrada como mouse e teclado sem depender de uma plataforma específica, estamos utilizando a OIS (Object-Oriented Input System) que é compatível com a OGRE.

Basicamente o sistema é composto por classes que geram e manipulam a interface gráfica e por classes que efetuam a comunicação com o sistema de arquivos. A interação entre estes dois permite que seja possível representar objetos do sistema de arquivos em uma nova metáfora. O diagrama de classes da figura 2 apresenta as principais classes do sistema, as quais são descritas a seguir:

- *Application*: é a principal classe da interface gráfica. Utilizando o framework disponibilizado pela OGRE, é criado o gerenciador de cenas, é definido como será o ambiente, as câmeras de visualização e os objetos da interface como botões e janelas. Esta classe também inicializa a classe Listener e as *threads* da classe FileManagement para que seja possível a comunicação com o sistema de arquivos.



**Figura 2. Diagrama de classes do sistema**

- *Listener*: classe responsável pela manipulação dos eventos da OIS e a renderização das modificações que estes eventos produzem no ambiente. Qualquer evento do mouse, teclado ou do sistema de arquivos gera uma chamada nos métodos desta classe para que seja devidamente tratado no ambiente.
- *GUIEvents*: esta classe declara quais métodos são utilizados quando eventos da CEGUI (Crazy Eddie's GUI System) são executados, como pressionar um botão e mover uma janela.
- *Factory*: classe é utilizada para abstrair a gerência dos objetos na interface. Consiste em métodos para criar, localizar, excluir, alterar formato dos objetos e associação com objetos do sistema de arquivos.
- *FileManagement*: a comunicação com o sistema de arquivos é realizada por esta classe. Seja recebendo notificações de alterações pela API ou efetuando alterações em arquivos e diretórios de acordo com as modificações realizadas pelos objetos da interface. O monitoramento das alterações é realizado através da API WIN 32 (IVANOV,2007; HALERSHTEIN,2007) a qual disponibiliza um conjunto de funções (FindFirstChangeNotification, FindNextChangeNotification, FindCloseChangeNotification) para configurar a abrangência das notificações a serem consideradas. Dentre o conjunto de atividades que podem ser monitorados podem ser citados: atividades de criação, exclusão e alteração de nome de arquivos e diretórios; (ii) a alteração de tamanho de arquivos; (iii) alteração da hora de gravação dos arquivos; e (iv) alterações nos descritores de segurança.

A figura 3 apresenta a versão atual do modelo de interface sendo construído. Como pode ser observado, embora ainda esteja em uma fase inicial, o projeto de interface está baseado nos conceitos desenvolvidos no projeto Data Mountain apresentado anteriormente.

#### 4.1. Implementação

Conforme descrito na arquitetura, a ferramenta se baseia no motor gráfico OGRE e seu framework, vinculando a interface gráfica ao sistema de arquivos. O framework do OGRE disponibiliza classes para criar a interface, classes para renderização e é compatível com a CEGUI e OIS, que disponibilizam as classes para criar a interface e tratar eventos de mouse e teclado.

A classe *Application* é essencial em uma aplicação que utiliza a engine OGRE. Em seu corpo está o método `createScene` (quadro 1). Este método cria as principais

configurações da interface, como definir a iluminação e carregar o arquivo que contém as configurações de terreno, tamanho da área disponível, etc. Além disso, efetua a integração com a CEGUI e inicializa a thread que irá monitorar os arquivos. Neste método também é efetuada a chamada do método loadPersistentObjects, que irá carregar os objetos criados anteriormente pelo usuário. O método loadPersistentObjects consiste em percorrer os arquivos de cada objeto, definir sua posição na tela e seus arquivos/diretórios relacionados, para que os mesmos sejam posteriormente integrados com a classe FileManagement que irá efetuar a monitoração.

#### Quadro 1 – Método createScene

```
void createScene(void) {  
    // configuração da luz do ambiente  
    mSceneMgr->setAmbientLight(ColourValue(0.5, 0.5, 0.5));  
    mSceneMgr->setSkyDome(true, "Examples/CloudySky", 5, 8);  
    // carrega o arquivo que contém as definições do mundo  
    mSceneMgr->setWorldGeometry( "terrain.cfg" );  
    // configuração da GUI  
    mGUIRenderer = new CEGUI::OgreCEGUIRenderer(mWindow,  
        Ogre::RENDER_QUEUE_OVERLAY, false, 3000, mSceneMgr);  
    // cria a thread que irá monitorar os arquivos  
    hFileSystemListener = CreateThread(NULL,0,  
        (LPTHREAD_START_ROUTINE)FileSystemListener,NULL,0,&iID);
```

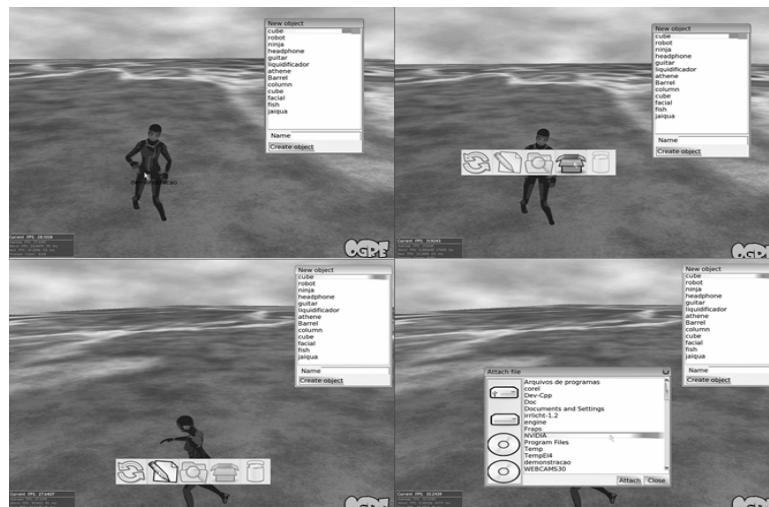


Figura 3 – Exemplo do modelo de interface

## 5. Conclusões

O presente trabalho descreveu o estado atual da concepção de um sistema de visualização de arquivos em um sistema de arquivos utilizando uma metáfora 3D. Os fundamentos que sustentam o projeto foram apresentados e a arquitetura do sistema é descrita.

Os aspectos de visualização 3D propostos na ferramenta levam em consideração as características da informação armazenada no disco, uma vez que o usuário é capaz de definir quais objetos irão representar seus conjuntos de dados. Ou seja, é possível definir que informações com características semelhantes serão representadas por objetos que mais se aproximem visualmente destas características. No estágio atual da implementação estão sendo construídos os objetos gráficos que permitem esta associação. Nesta fase do projeto ainda não é possível apresentar um comparativo de

performance e usabilidade com relação aos projetos descritos nas seções 2 e 3. Contudo, pretende-se avaliar a usabilidade da ferramenta utilizando a mesma metodologia adotada no projeto Data Mountain. Apesar disto, é possível destacar que a metáfora utilizada difere daquelas apresentadas anteriormente na medida em que está utilizando o conceito de avatares que se movem num universo virtual ao invés de utilizar representações icônicas dos arquivos e pastas de um sistema de arquivos. Este modelo de interface abre a possibilidade de investigação na incorporação de inteligência no avatar de forma a permitir que o mesmo possa aprender as preferências do usuário a partir do uso. O projeto está previsto para encerrar em junho de 2007 quando deverá ser apresentado como trabalho de conclusão de curso.

## Referências

- AGARAWALA, A.; BALAKRISHNAN, R. **Keepin' it real**: pushing the desktop metaphor with physics, piles and the pen. Toronto, 2006. Disponível em: <<http://honeybrown.ca/Pubs/BumpTop.pdf>>. Acesso em: 30 ago. 2006.
- APPERLEY, Mark D.; TZAVARAS, Yannis; SPENCE, Robert. A Bifocal display technique for data presentation. In: CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS., 2., 1982, Manchester. **Proceedings...** Manchester: North-Holland Publishing Company, 1982. p. 27-43.
- BEDERSON, B. **Fisheye Menus**. Proceedings of ACM Conference on User Interface Software and Technology (UIST 2000), pp. 217-226. ACM Press. 2000.
- COCKBURN, A.; MCKENZIE, B. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 22., 2002, Minneapolis. **Proceedings...** Alpha: Sheridan Printing, 2001. p. 138-146.
- FARHOOOMAND, A. F.; DRURY, D.H. Managerial information overload. **Communications of the ACM**. Nova Iorque, v. 45, n. 10. p. 22-36. Out. 2002.
- FREITAS, C. M. et al. Introdução à visualização de informações. **Revista de Informática Teórica e Aplicada**, Porto Alegre, v. VIII, n. 2. p. 143-158. Out. 2001.
- HALERSHTAIN, Vitali . Spying a file system.[S.I.], 2007. Disponível em: <<http://www.codeproject.com/file/FileSpyArticle.asp?print=true>>. Acesso em: 03 mar 2007.
- IVANOV, Ivo. API hooking revealed. [S.I.], 2007. Disponível em: <<http://www.codeproject.com/system/hooksys.asp?print=true>>. Acesso em: 10 mar 2007.
- MACKINLAY, J. D. Et al. The perspective wall: detail and context smoothly integrated. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 11., 1991, New Orleans. **Proceedings...** New Orleans: [s.n.], p. 173-176. 1991.
- ROBERTSON, George et al. **Data Mountain**: using spatial memory for document management. Proceedings of the 11th annual ACM Symposium on User Interface Software and Technology. San Francisco, Califórnia, 1998. Disponível em: <<http://www.microsoft.com/usability/UEPostings/p153-robertson.pdf>>. Acesso em: 15 ago. 2006.
- ROBERTSON, George et al. **Task Galley**: a 3D window manager. Vermont, 1999. Disponível em: <<http://research.microsoft.com/~marycz/tg2000.pdf>>. Acesso em: 20 set. 2006.
- SHINYAMA, Y. **XCrusier**. [S.I.], 2003. Disponível em: <<http://xcrusier.sourceforge.net>>. Acesso em: 20 set. 2006.
- TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas operacionais**: projeto e implementação. 2. ed. Tradução Edson Furmarkiewicz. Porto Alegre: Bookman, 2000.
- UNIVERSITY OF BRISTOL. **Neural basis of spatial memory**. Bristol, [2001?]. Disponível em: <<http://www.bris.ac.uk/depts/Synaptic/research/projects/memory/spatialmem.htm>>. Acesso em: 28 ago. 2006.