

# PREGO-Um Editor Gráfico de Objetos Procedurais

NEIL ARMSTRONG REZENDE<sup>1</sup>

WALDEMAR CELES FILHO<sup>1</sup>

MARCELO GATTASS<sup>1</sup>

ENIO EMANUEL RAMOS RUSSO<sup>1,2</sup>

<sup>1</sup>TeCGraf-Grupo de Tecnologia em Computação Gráfica  
Departamento de Informática, PUC-Rio  
Rua Marquês de São Vicente, 255 – 22453-900 Rio de Janeiro, RJ, Brasil  
{neil, celes, gattass}@icad.puc-rio.br

<sup>2</sup>PETROBRAS-Petróleo Brasileiro S.A.  
CENPES-Centro de Pesquisas e Desenvolvimento Leopoldo A. Miguez de Mello  
Cidade Universitária, quadra 7, sala 9020 – 21949-900 Rio de Janeiro, RJ, Brasil  
e210@c53000.petrobras.anrj.br

**Abstract.** We describe an interactive drawing editor that works with graphical and textual views. This approach allows the user to create and edit procedural objects.

**Keywords:** Computer Graphics, Graphical Editor, Programming Language.

## 1 Introdução

Editores gráficos podem ser avaliados segundo dois fatores: flexibilidade e precisão. Por flexibilidade, entendemos que um editor possa ser utilizado para a criação de diversos tipos de representações gráficas, sejam elas de cunho técnico-científico ou artístico. O fator precisão está relacionado com a exatidão na definição das representações gráficas. Este fator é imprescindível para atender aplicações técnico-científicas, onde precisamos criar desenhos em escalas precisas. Aliado a estes fatores, deve-se considerar a facilidade com que os desenhos são criados, utilizando técnicas apropriadas de interface com o usuário.

Os editores gráficos convencionais não provêm recursos adequados para se ter flexibilidade e precisão, sem perder a característica de ser fácil para o usuário criar suas representações. Não é comum esses editores permitirem a instanciação de objetos complexos, tais como representação de funções matemáticas ou de fractais. Para prover precisão na criação dos desenhos, alguns editores, como o AutoCAD, permitem a especificação de comandos via teclado. Esta capacidade, entretanto, ou está limitada aos comandos pré-definidos atuando sobre primitivas básicas, ou sobre primitivas previamente definidas através de uma linguagem de configuração.

É interessante acrescentar a um editor gráfico o poder de expressão que encontramos nas linguagens procedurais. Este tipo de abordagem é encontrado somente em alguns editores de animação gráfica, onde encontramos

uma linguagem de manipulação de objetos gráficos, chamados de “atores” [1].

Neste trabalho, apresentamos um editor gráfico que permite a edição do desenho via uma linguagem procedural, além de oferecer uma interface de manipulação direta convencional. O editor apresenta uma interface híbrida [2,5], onde cada objeto pode ser representado gráfica ou textualmente. Com isto, o usuário tem acesso às operações gráficas usuais e pode programar ações acessando uma linguagem procedural acoplada com a representação textual.

## 2 Arquitetura

A arquitetura do editor gráfico proposto é ilustrada pela Figura 1. O editor é implementado sobre uma biblioteca de classes para armazenamento estruturado e edição de uma estrutura hierárquica de primitivas gráficas, chamada GLB, e acopla a linguagem de programação Lua [3,4]. O editor gráfico trabalha tanto sobre representações gráficas dos objetos da GLB quanto sobre representações textuais associadas a eles. A linguagem Lua é responsável por interpretar as edições feitas na representação textual e atualizar a biblioteca GLB.

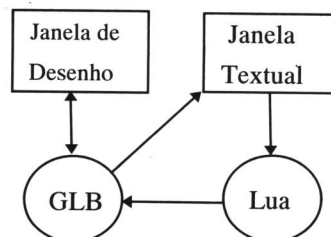


Figura 1: Arquitetura do editor gráfico.

### A biblioteca GLB

A GLB é implementada sob o paradigma de orientação a objetos, e possui classes que associam grupos hierárquicos de primitivas a áreas de desenho. O editor PREGO utiliza os serviços da GLB para editar esta estrutura e age diretamente sobre a área de desenho. Os objetos gráficos armazenados na GLB podem ser apresentados na janela textual, bem como serem gravados em um arquivo no formato Lua.

### A linguagem procedural

Lua é uma linguagem de extensão projetada para ser usada como linguagem de configuração, acoplada a um programa hospedeiro. Aplicações em geral podem acoplar códigos em Lua, permitindo prototipagem rápida e acesso programável pelo usuário à tecnologia implementada pela aplicação. Para o editor PREGO, criamos uma especialização de Lua para instanciação e manipulação dos objetos gráficos representados pela GLB. O editor pode chamar funções em Lua para serem executadas, bem como ler e atribuir variáveis de Lua. Além disso, Lua provê as construções fundamentais para a definição de controle de fluxo (*if*, *while*, *repeat*) e um poderoso mecanismo para descrição de objetos.

### 3 Funcionalidade

A Figura 2 ilustra a interface híbrida do editor. O usuário pode criar, alterar ou agrupar as primitivas do desenho por manipulação direta da maneira convencional ou através de codificação em Lua. Dado um conjunto de primitivas representadas graficamente no *canvas*, podemos ter uma descrição textual destas na área de texto. A cada objeto gráfico, associa-se um nome simbólico para que possa ser referenciado na área textual. A descrição textual é interpretada por Lua que atualiza a área gráfica através da GLB, conforme discutido na seção anterior.

Uma das principais características deste sistema é a possibilidade de criação e manipulação de novas primitivas segundo a descrição destas através de uma função em Lua. Desta forma, primitivas mais complexas, tipo o gráfico de uma função matemática ou a definição de uma fractal, podem ser também definidas e tratados como objetos vetoriais.

### 4 Conclusão

A abordagem de duas visões para manipulação de objetos já foi usada na construção de interfaces gráficas[5] baseadas em elementos de interface simples e pré-definidos. Neste trabalho, estende-se estes conceitos para elementos de dados genéricos tratados proceduralmente. A utilização de uma linguagem

procedural na definição e manipulação de objetos confere uma grande flexibilidade aos editores gráficos, e permite a instanciação de objetos em escalas precisas.

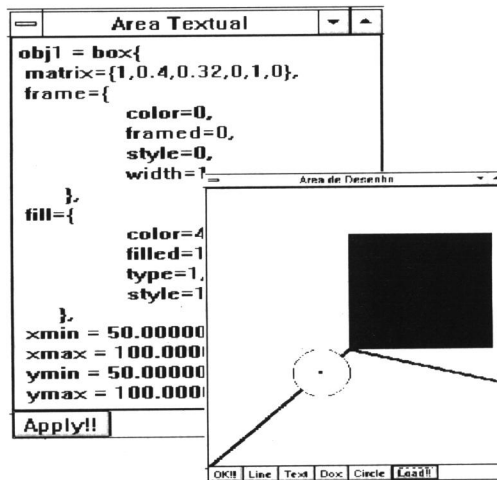


Figura 2: Interface gráfica e textual.

A possibilidade do usuário programar ações sobre os objetos instanciados, combinada com os recursos de manipulação direta dos editores convencionais, permite que tarefas de interação complexas e trabalhosas sejam facilmente realizadas.

O poder de expressão de linguagens procedural é explorada para permitir a instanciação e manipulação de objetos procedurais complexos, tais como funções e fractais.

*Agradecimentos:* Este trabalho teve suporte parcial do projeto temático do CNPq GEOTEC - Geoprocessamento: Sistemas e Técnicas.

### Referências

- [1] C. W Reynolds, "Computer animation with scripts and actors", *Proc SIGGRAPH '82, Computer Graphics*, 16(3), 25-34, 1982.
- [2] G. Aurahans, K. P. Brooks, M. H. Brown, "A Two-View Approach to Constructing User Interfaces", *ACM - Computer Graphics*, 23(3), July 1989.
- [3] L. H. de Figueiredo, R. Ierusalimschy e W. Celes Filho, "The design and implementation of a language for extending applications", *Anais do XXI Semish*, 273-283, 1994.
- [4] R. Ierusalimschy, L. H. de Figueiredo e W. Celes Filho, "Reference manual of the programming language Lua 2.1", *Monografias em Ciência da Computação*, Dep. de Informática, PUC-Rio, 1995.
- [5] R. O. Prates, "Visual LED: Uma ferramenta Interativa para Geração de Interfaces Gráficas", *PUC-RJ - Dissertação de Mestrado*, Departamento de Informática, PUC-Rio, 1994.