

# Editor Caligráfico para Gestão de Ambientes Virtuais

Alfredo Ferreira Marco Vala Guilherme Raimundo Joaquim Jorge Ana Paiva J.A.Madeiras Pereira

Departamento de Engenharia Informática

INESC-ID/IST/Universidade Técnica de Lisboa

alfredo.ferreira@inesc-id.pt, {marco.vala,guilherme.raimundo}@tagus.ist.utl.pt, jorgej@acm.org, {ana.paiva,jap}@inesc-id.pt

## Resumo

A simulação de mundos reais através de Ambientes Virtuais (AV) é uma área em crescimento com um vasto leque de aplicações práticas. Actualmente existem diversas ferramentas para criar e visualizar AVs que recorrem a interfaces gráficas, mas estas são geralmente de funcionalidade e usabilidade limitadas. No seguimento dos recentes desenvolvimentos em sistemas caligráficos, concebemos uma abordagem que tira partido deste tipo de sistemas para gerir um AV. Nesta comunicação apresentamos uma solução que permite aos utilizadores gerirem um AV de forma colaborativa usando uma interface caligráfica.

## Palavras-Chave

Interfaces caligráficas, Interfaces Colaborativas, Ambientes Virtuais

## 1 Introdução

O mundo real pode ser visto como um conjunto de entidades dinâmicas, que interagem entre si, criando assim um universo em constante mutação. Nos últimos anos temos assistido a uma necessidade cada vez maior de simular esta dinâmica em mundos virtuais. Seja para estudar o comportamento de uma multidão durante a evacuação de um edifício em chamas, seja para incorporar um herói numa aventura para fins lúdicos. No entanto, para se atingirem os objectivos pretendidos é necessário visualizar e manipular o ambiente virtual (AV) associado à simulação.

Na abordagem proposta usamos a *framework* ION para realizar as simulações em AVs, tendo esta a responsabilidade de garantir que as regras do mundo são respeitadas. Para manipular e representar simbolicamente os elementos destes AVs, desenvolvemos uma ferramenta que permite gerir de forma colaborativa um AV antes e durante a simulação. Inicialmente apresentado em [Ferreira 06], o EditION usa uma interface caligráfica de modo a tirar partido da forma como os criadores de AVs os desenham em folhas de papel antes de os criarem no sistema. Recorrendo a uma linguagem visual simples, esta ferramenta permite aos seus utilizadores criarem mundos virtuais através de esboços desenhados directamente no computador, reduzindo assim a necessidade de codificação usualmente associada a estas tarefas.

## 2 Arquitectura

Apesar de o EditION estar intimamente ligado à *framework* ION, estes são completamente independentes. Na realidade, é o núcleo da *framework* que suporta o ambiente virtual, fornecendo uma API para controlo externo. Este controlo pode ser realizado remotamente por diver-

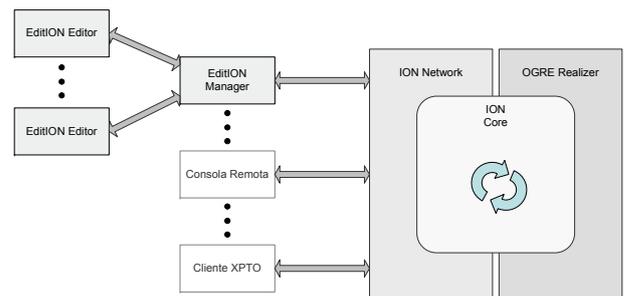


Figura 1. Arquitectura

sas aplicações, desde uma simples consola textual até ao nosso editor caligráfico. Na Figura 1 ilustramos a arquitectura da solução proposta. A simulação é de facto executada no módulo ION.Core, enquanto que o módulo ION.Network fornece pontos de acesso às funcionalidades do ION.Core a clientes externos, eventualmente remotos. Um destes clientes é o EditION.Manager que gere a informação visual da representação simbólica, oferecendo o suporte necessário ao funcionamento distribuído do EditION.Editor. Por outro lado, o módulo OGRE Realizer permite a visualização tri-dimensional do mundo virtual.

## 3 Editar e Depurar com o EDITION

Com esta arquitectura os utilizadores do EditION podem desenhar os AVs usando uma caneta, reduzindo a demorada e cansativa tarefa de escrita de inúmeras linhas de código. Basta esboçar os elementos do mundo virtual que estes são automaticamente criado na *framework* ION.

Para efectuar o reconhecimento de gestos recorreremos à

biblioteca CALI [Fonseca 02]. Como este reconhecimento é imediato, o esboço é interpretado e validado, quer sintática quer semanticamente, enquanto está a ser desenhado. Assim não existe necessidade de desenhar todo o mundo virtual e codificá-lo no sistema para verificar a existência de erros, como geralmente acontece com outras ferramentas. Por outro lado, a vertente colaborativa desta solução garante que as alterações realizadas no mundo virtual por um utilizador são imediatamente visualizadas pelos restantes.

Para além de criar e alterar elementos do mundo virtual, o utilizador pode também esboçar comandos através de gestos pré-definidos. Estes permitem controlar o fluxo de execução da simulação e manipular os atributos dos elementos do mundo. Desta forma o utilizador consegue depurar a simulação em tempo real eficazmente.

A solução aqui apresentada foi testada num caso de estudo baseado no *FearNot!* [Aylett 05], uma aplicação desenvolvida para simular o *bullying* nas escolas. Através da utilização de duas instâncias do *EditION.Editor*, foi criado na *framework* ION o mundo onde iria decorrer a simulação. Uma vista parcial da representação simbólica desse mundo no editor está ilustrada na Figura 2. Aqui podemos ver representadas duas entidades, o atacante e a vítima, e as respectivas propriedades e acções. Durante o decorrer da simulação estas foram manipuladas em paralelo por dois utilizadores, em computadores distintos de forma colaborativa, afectando o comportamento dos personagens.

Em simultâneo, enquanto a simulação era executada na *framework* ION e manipulada nas duas instâncias do *EditION.Editor*, o módulo *OGRE Realizer* fornecia uma visualização tri-dimensional do mundo, conforme ilustrado na Figura 3. Esta vista, no entanto, não possui a informação necessária para uma análise detalhada da interacção entre os personagens, oferecida pela representação simbólica disponível no editor.

#### 4 Conclusões e Trabalho Futuro

Apresentámos aqui uma solução colaborativa para gerir AVs tirando partido de interfaces caligráficas. Usando

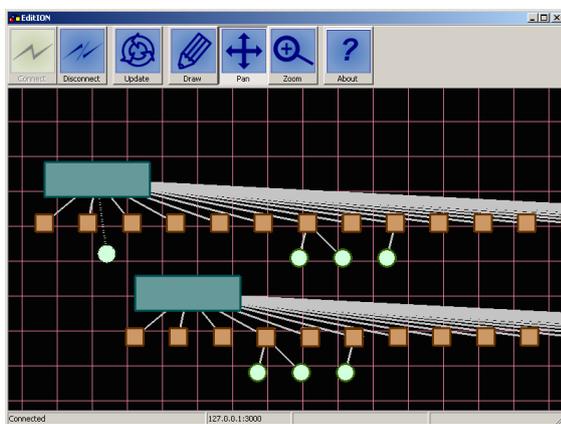


Figura 2. Vista do mundo no *EditION*

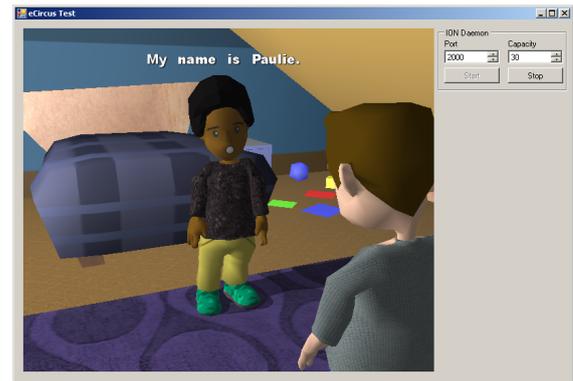


Figura 3. Vista do mundo no *OGRE Realizer*

esboços, semelhantes aos desenhados numa folha de papel, o utilizador pode criar, editar e controlar AVs. No entanto, existe ainda lugar para melhoramentos. A dinâmica inerente às simulações levanta questões relativas à organização espacial automática dos elementos na representação simbólica. Por outro lado a linguagem visual pode ser estendida de forma a enriquecer a visualização. Concluindo, julgamos que esta solução fornece novas contribuições à área da simulação e constitui um primeiro passo no desenvolvimento das ferramentas do futuro para a gestão de ambientes virtuais.

#### 5 AGRADECIMENTOS

Este trabalho foi parcialmente patrocinado pela Fundação para a Ciência e Tecnologia (bolsas SFRH/BD/17705/2004 e SFRH/BD/25725/2005) e pelo projecto eCIRCUS (contrato IST-4-027656-STP) parte do VI programa quadro da Comunidade Europeia.

#### Referências

- [Aylett 05] R. Aylett, S. Louchart, J. Dias, A. Paiva, e M. Vala. *Fearnot! - an experiment in emergent narrative*. Em *Proceedings of Fifth International Working Conference on Intelligent Virtual Agents, IVA 2005*, páginas 305–316, 2005.
- [Ferreira 06] Alfredo Ferreira, Marco Vala, J.A. Madeiras Pereira, Joaquim A. Jorge, e Ana Paiva. A calligraphic interface for managing agents. Em *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2006*.
- [Fonseca 02] Manuel J. Fonseca, César Pimentel, e Joaquim A. Jorge. CALI: An Online Scribble Recognizer for Calligraphic Interfaces. Em *Proceedings of the 2002 AAI Spring Symposium - Sketch Understanding*, páginas 51–58, Palo Alto, USA, Março 2002.