

# PROTOTIPAGEM RÁPIDA DE AMBIENTES DE HIPER- REALIDADE VIRTUAL

Carlos Martinho<sup>1</sup>  
José António Proença<sup>1</sup>  
Tiago Sepúlveda<sup>1</sup>  
Mário Rui Gomes<sup>1,2</sup>

<sup>1</sup>INESC, R. Alves Redol 9, 1000 LISBOA, Portugal

<sup>2</sup>IST, Av. Rovisco Pais 4, 1000 LISBOA, Portugal

## Sumário

Neste artigo é descrito o *GAVE*, um módulo que proporciona a navegação em espaços de *hiper-realidade* virtual construídos semi-automaticamente. Por *hiper-realidade* virtual entende-se uma rede de espaços tridimensionais interligados e enriquecidos por um conjunto de eventos audiovisuais 3D navegáveis através de uma *interface* imersiva. No artigo são revistos os conceitos fundamentais do *GAVE* e são descritas duas aplicações efectuadas com este módulo.

## 0. Introdução

Uma das maiores dificuldades no desenvolvimento de aplicações de Realidade Virtual (RV) de uma forma massiva prende-se com o facto de requerer conhecimentos específicos, fora do alcance de qualquer pessoa que não tenha um nível de formação razoável em programação avançada. A nosso ver, o acesso a ferramentas de visualização tão poderosas como são as aplicações tridimensionais interactivas em tempo real, deve ser facilitado por forma a que possam ser utilizadas em contextos mais alargados onde, muitas vezes, o recurso a profissionais de informática não é possível. A junção deste objectivo com o desejo dos autores de desenvolver uma ferramenta de prototipagem expedita de ambientes virtuais produziu o *GAVE*, *Generic Abstraction for Virtual Environments*, um módulo que estabelece a ligação, de um modo imediato e acessível, entre um programa de modelação de uso comum, como o *3D Studio MAX* da *Kinetix* e uma API destinada à elaboração de aplicações de RV, no caso o *RenderWare* da *Criterion Software Limited*. A utilização do *3D Studio MAX* traz vantagens determinantes como a rápida visualização dos elementos num ambiente extremamente potente e especializado na modelação e visualização tridimensional.

Para desenvolver um ambiente virtual interactivo [Kal94][Fol92], usando o *GAVE*, um utilizador constrói a base de dados geométrica da cena pretendida e enriquece-a em objectos

que consubstanciam mecanismos genéricos como colisões, áreas de activação de eventos ou iluminação. Em seguida, o preenchimento de um ficheiro de texto configura todo o sistema de uma forma transparente. Este ficheiro de configuração é o cerne de todo o *GAVE*, já que neste estão descritas todas as propriedades do sistema, entre outras, dispositivos de entrada, tipos de câmara, segmentos de colisão, áreas de activação de eventos, *set-up* de eventos.

A invocação do núcleo do *GAVE*, parametrizada pelos ficheiros de descrição da cena e de configuração do ambiente sintético, instancia a aplicação de RV imaginada pelo utilizador.

## 1. Contexto

A maioria dos *toolkits* destinados à construção de ambientes virtuais, tais como o *WorldToolkit* ou o *Performer* [Kal94][Perfo], são bibliotecas de algoritmos específico, inutilizáveis para quem não tenha conhecimentos profundos de programação. São apenas colecções de funções, muitas vezes sem qualquer suporte abstracional e conceptual estruturante.

Ao longo dos anos, alguns destes motores têm vindo a ser enriquecidos com ferramentas de visualização e/ou modelação [Sense]. No entanto, a maioria destas ferramentas não facultam os mecanismos normalmente associadas a aplicações de RV, como por exemplo, a introdução de colisões.

Algumas aplicações como o *Sim Studio* [Ndime] permitem ainda a configuração do comportamento de alguns objectos segundo uma aproximação orientada ao Agente.

Existem porém, algumas excepções (recentes) de ferramentas visuais de construção de ambientes virtuais que já facultam a introdução de características tradicionais nos sistemas de RV, tal como aliás acontece no *GAVE*. Por exemplo, o *VREK* da *ThemeKit* [Theme], baseado em MS-DOS, é constituído por uma série de módulos que permitem a construção de um ambiente virtual desde a modelação de objectos até ao posicionamento de objectos na cena, iluminação ou estabelecimento de colisões.

Apesar de a *interface* de aplicações como o *VREK* serem geralmente desenhadas de forma a permitir uma fácil apreensão de conceitos, um utilizador que não tenha conhecimentos aprofundados de informática prescindirá sempre de aprender a trabalhar com uma nova ferramenta, se essa ferramenta não lhe fornecer serviços que ele considere absolutamente necessários, tais como a visualização interactiva em tempo real. O *GAVE* neste aspecto apresenta características únicas, já que reutiliza uma interface já familiar para criar e configurar o ambiente sintético que se pretende visualizar.

## 2. Especificação

### 2.1 Arquitectura

O *GAVE* é uma camada de abstracção, programada sobre a API *Renderware* [Fer95], que estabelece toda a semântica de conceitos necessária à construção automática de espaços de *hiper-realidade* virtual, isto é, uma rede de espaços tridimensionais interligados e enriquecidos por um conjunto de eventos audiovisuais 3D inspirada no conceito de hiper-média [Nie94].

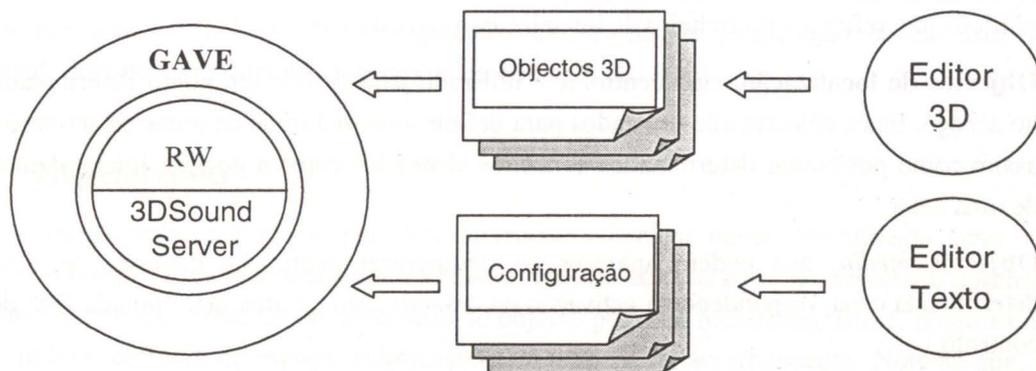


Fig. 1 - Arquitectura do *GAVE*.

O *GAVE* baseia-se num motor de estados configurado à partida com um ficheiro exterior. Este motor recebe dados do utilizador para progredir nos estados. A máquina de estados actua na base de dados 3D e conseqüentemente altera a sua visualização. As funcionalidades do *GAVE* podem ser ligadas à base de dados tridimensional e à máquina de estados segundo a configuração dada pelo utilizador. A flexibilidade desta ligação é uma mais valia do *GAVE*. A máquina de estados é constituída por muitas outras máquinas de estado independentes e que constroem todas as funcionalidades do *GAVE*.

O *GAVE* corre em plataforma PC e está, de momento, preparado para interagir com os *i-glasses* e com um conjunto de dispositivos sem fio, para além dos tradicionais *joysticks*. A implementação do som efectua-se através da ligação ao *3DSoundServer* [Pro97], um servidor de som localizado tridimensionalmente.

### 2.2 Utilização

O utilizador começa por definir um conjunto de espaços virtuais através de uma ferramenta gráfica tridimensional como, por exemplo, o *3D Studio Max* [Ell96]. Em cada espaço, o utilizador adiciona, no próprio ambiente 3D, o conjunto dos objectos de parametrização através dum simbolismo visual que ele próprio define e que especifica no ficheiro onde está também especificada a parte funcional do espaço assim como a configuração dos periféricos utilizados para a interacção.

Depois, basta executar o núcleo *GAVE* o qual gere a navegação, a interacção, os eventos e a realimentação audiovisual de acordo com as especificações do utilizador.

### 2.3 Conceitos

A unidade fundamental do *GAVE* é a **cena**. A cena é uma unidade abstracta constituída por um conjunto de objectos tridimensionais que podem ser de três tipos:

- **Objectos estáticos**, que são visualizados em simultâneo quando a cena é carregada e desvanecem também simultaneamente quando esta desaparece. Todos os objectos gráficos não referidos no ficheiro de inicialização são estáticos.
- **Objectos de localização**, cujo centroide é utilizado para definir um ponto determinado no espaço. Estes objectos são utilizados para definir uma multitude de áreas de activação assim como posicionar determinados conceitos abstractos como a posição inicial dentro de uma cena.
- **Objectos *media***, que podem aparecer ou desaparecer dentro do contexto de uma determinada cena, respondendo à activação ou desactivação de uma determinada área de controlo.

Note-se que todo o objecto gráfico pertence a uma cena e as suas propriedades só estão activadas enquanto a cena está activa. O que poderia parecer uma séria limitação não o é de facto uma vez que a reutilização de objectos pode ser obtida pelo uso de referências a um mesmo ficheiro gráfico em cada cena, sendo cada instância um objecto distinto.

#### 2.3.1 Objectos estáticos

Apesar de todos os objectos gráficos não referidos no ficheiro de configuração serem objectos estáticos, estes podem ser parametrizados e assim aparecerem nesse mesmo ficheiro. A opacidade do objecto, a aplicação de máscaras de opacidade e a orientação dinâmica para o utilizador são as propriedades configuráveis.

#### 2.3.2 Objectos de localização

Os objectos de localização são utilizados apenas para identificar uma determinada posição dentro da cena. Não são visualizados pelo utilizador e por isso a definição do formalismo visual a que o utilizador recorre está inteiramente ao seu critério. Quando um conceito abstracto é definido no ficheiro de configuração, os objectos de localização servem de parametrização desse conceito. Podem ser utilizados individualmente, para indicar posições de salto (por exemplo onde o utilizador deve aparecer depois da ocorrência de um determinado evento ter provocado a mudança de cena. Utilizando objectos de localização dois a dois, é possível definir segmentos de colisão que nunca poderão ser atravessados pelo

utilizador e cuja intersecção poderá lançar um determinado som. Utilizando três ou mais objectos de localização, é possível definir um polígono que representa uma área de activação, isto é, uma área à qual é possível atribuir uma determinada funcionalidade mediática: *hotspot* ou *portal*. A entrada num determinado *hotspot* activa um determinado objecto *media* e a respectiva saída desactiva-o. A entrada na área de um portal inicia uma transição de cena, que corresponde a um *cross-fade* parametrizado entre duas cenas, com um possível acompanhamento de eventos sonoros, passíveis de um efeito análogo.

Os objectos de localização são também utilizados para definir a posição de pontos de iluminação, que actualmente são perdidos na exportação do ficheiro e assim devem ser parametrizados no ficheiro de configuração. Um ponto de localização define uma luz pontual, dois definem uma luz direccional ou cónica.

### 2.3.3 Objectos *media*

Um objecto *media* é activado por um determinado *hotspot* numa determinada cena. Ao activar o *hotspot* é iniciado um *fade-in* visual do objecto ao qual pode ser associado o *fade-in* de um determinado som. O som associado ao objecto pode ser localizado, isto é, possuir uma origem bem definida no espaço tridimensional e perceptível ao *cibernauta*. Note-se que, se todo o som tem de estar associado obrigatoriamente a um objecto *media*, no entanto, este não tem de possuir obrigatoriamente uma representação material, isto é, um *hotspot* pode activar apenas som. Ao sair do *hotspot* de activação é iniciado um *fade-out* visual, acompanhado ou não por um *fade-out* áudio associado. Note-se que todos os objectos são independentes e que pode acontecer verificar-se *cross-fades* entre objectos *media* distintos, basta atender, por exemplo, a dois *hotspots* contíguos atribuídos a dois objectos *media* muito próximos ou mesmo sobrepostos.

Como nota final, a localização do *hotspot* não tem qualquer relação com a localização do objecto *media*, sendo possível activar objectos fora do alcance visual do *cibernauta*.

Para além de toda a configuração do espaço de *hiper-realidade* virtual, o ficheiro de configuração permite ainda a configuração dos dispositivos de entrada e saída da aplicação, desde a sua sensibilidade aos pormenores mais íntimos de forma a ser possível adaptar a aplicação a outros componentes *hardware* de forma fácil e sem custos adicionais.

## 3. Aplicações

O *GAVE* foi utilizado na produção de duas aplicações de RV que foram exibidas ao público: *Salas da História* e *KM*.

### 3.1 Salas da História

O *Salas da História* foi a primeira aplicação concebida com o *GAVE*. Surgiu de uma

solicitação do Centro Multimédia do INESC para a elaboração de uma aplicação de RV que demonstrasse o potencial desta tecnologia para fins educativos. O *Salas da História* foi exposto na última semana de Outubro de 1996 na Sala do Futuro do Centro de Informação Jacques Delors, sendo posteriormente exibido no 3º Fórum Emprego e Formação (FormE) que decorreu na Exponor de 2 a 6 de Novembro de 1996, organizado pelo Instituto de Emprego e Formação Profissional.

### 3.1.1 Conceito

Apesar de a aplicação ter como mote o Ensino e a Educação, o *Salas da História* foi concebido não como substituto mas como um complemento à acção pedagógica directa por parte do professor.

O conceito do *Salas da História* foi instanciado para a disciplina de História, dada a dificuldade com que os professores se deparam em arranjar elementos palpáveis ilustradores dos vários itens do currículo. O *Salas da História* supre esta carência ao levar o aluno numa viagem no tempo até uma época remota.

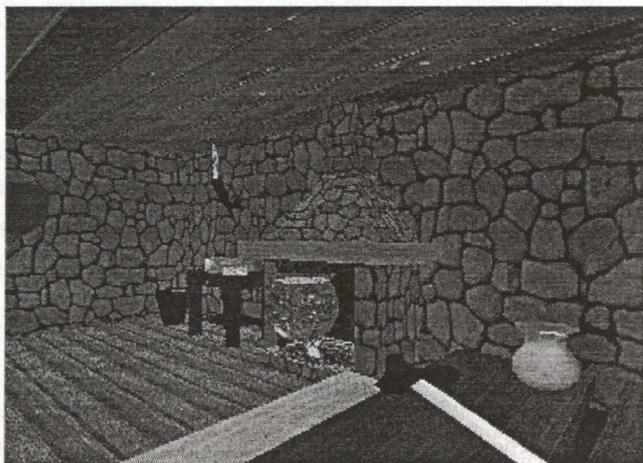


Fig. 2 - A cozinha medieval do *Salas da História*.

O ambiente desta aplicação é composto por duas cenas, que correspondem a duas cozinhas decoradas segundo duas épocas distintas: medieval e moderna. A disposição espacial das cozinhas é semelhante de forma a que o *cibernauta* possa identificar correctamente a função dos objectos expostos na cozinha medieval, apenas pela sua localização.

Em ambas as salas existem objectos específicos a que estão associados *hotspots* que activam eventos áudio (objectos *media*). Estes eventos consistem na locução de textos explicativos sobre os objectos em causa. Estes textos pretendem não só desvendar a função dos objectos escolhidos, mas também dar uma perspectiva histórica da evolução do dia-a-dia numa cozinha comum.

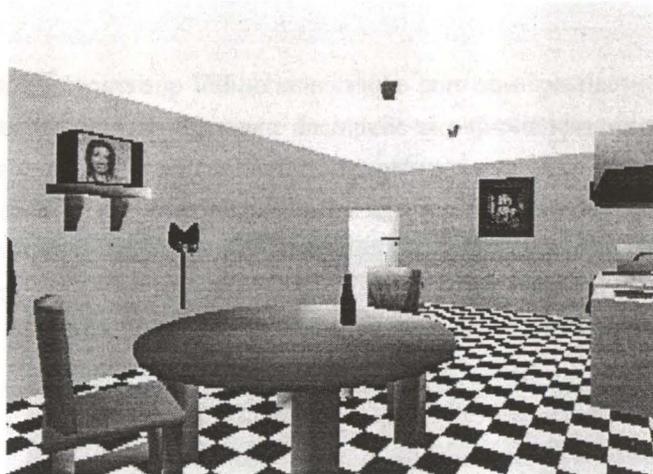


Fig. 3 - A cozinha moderna do *Salas da História*.

A passagem entre salas (cenas) implementada através de portais, sendo estes identificados por elementos decorativos deslocados da época a que a sala respeita. Na cozinha actual, o portal para a sala medieval é identificado por um machado. Na cozinha medieval a presença de um telefone indica a saída para a cozinha moderna.

### 3.1.2 Resultados

As duas salas existentes formam uma sessão educativa sobre História. Nestas salas, onde existem vários elementos didácticos, simulam-se ambientes de diferentes épocas, que vão proporcionando uma ilustração e explanação da vida daquelas épocas. Este conceito pode obviamente ser transposto para múltiplas salas e para os mais variados ambientes históricos.

O uso do *GAVE* facilitou a introdução dos elementos e mecanismos que constituem o *Salas da História*. A aplicação foi desenvolvida num espaço de tempo muito reduzido recorrendo única e exclusivamente ao ambiente de modelação e ao preenchimento dos ficheiros de configuração.

### 3.2 KM

O projecto *KM* encontra-se integrado na exposição *Ex Mater*, dinamizada pela Aula do Risco, cujo intuito é promover o futuro Parque Museu Virtual, a ser instalado em Montemor-o-Novo. *Ex Mater* esteve patente ao público no Museu Extremeño e Ibero-Americano de Arte Contemporânea de 4 de Setembro a 31 de Outubro de 1997, em Badajoz, Espanha. No início do ano de 1998, será exibida em Lisboa no Centro Cultural de Belém e em Maio passará para Montemor-o-Novo.

### 3.2.1 Conceito

O projecto consistiu na realização de uma experiência de RV que transmite ao utilizador um primeiro contacto com o ambiente que se viverá no museu do Parque Museu Virtual. Para este efeito, recorreu-se ao projecto arquitectónico deste espaço, no qual se recriou uma instalação, da autoria de António Cerveira Pinto, inspirada no erotismo e na sexualidade.

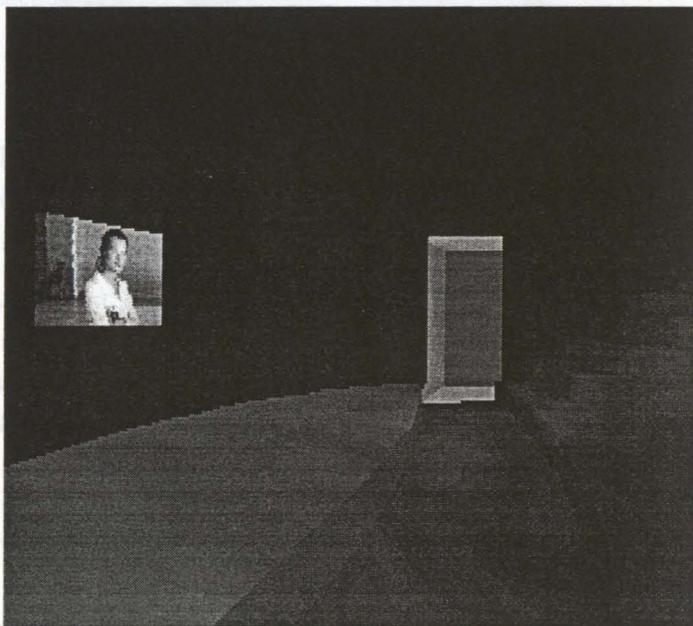


Fig. 4 - O auditório do futuro museu do Parque Museu Virtual.

A experiência é composta por duas salas de exposição e um auditório, nos quais se podem visualizar diversas fotografias e vídeos da modelo Kate Moss, elementos que representam o imaginário sexual masculino. As fotografias estão colocadas nas salas de exposição ao longo dos vidros exteriores e são visualizadas apenas quando o *cibernauta* penetra no *hotspot* correspondente. No ecrã do auditório está colocada uma fotografia deformada da modelo, que se transfigura num vídeo quando o *cibernauta* se aproxima do ecrã.

O imaginário feminino é concretizado através da audição de sons e música baseados no Canto de Corte de pássaros macho. Estes sons são despoletados através da passagem por áreas de activação colocadas aleatoriamente no espaço.

Todos os modelos tridimensionais foram executados no *3D Studio MAX* tendo por base a arquitectura das salas facultada pelo arquitecto autor do projecto. Depois de algumas simplificações e ajustes da arquitectura base, procedeu-se à criação dos elementos do interior das salas e também a experiências para recriar o céu e um espelho de água, que reflecte todo o ambiente e que circunda os vidros das salas.

### 3.2.2 Resultados

Sendo este um projecto com a ideia base criada por uma pessoa exterior à equipa de realização, foi de grande importância a faculdade do *GAVE* permitir praticamente sem custos visualizar e navegar nos ambientes protótipo que estávamos a modelar. Esta característica do *GAVE* permitiu uma grande interacção com o criador, particularmente importante por se tratarem de conceitos muitas vezes abstractos e de difícil concretização visual, intrinsecamente vinculados ao domínio artístico subjacente.

Este facto reforça ainda mais a convicção de que o processo criativo deve ser desempenhado pelo próprio conceptualizador do ambiente.



Fig. 5 - Com a aproximação a cada um dos vidros da *cibernauta* pode ver várias fotografias da modelo Kate Moss.

## 4. Futuro

Recentemente iniciou-se o desenvolvimento de uma nova versão do *GAVE* que vem na sequência da reformulação da abstracção e que inclui o alargamento da sua gama de funcionalidades com destaque para a possibilidade de se estabelecerem relações de causalidade entre eventos (incluindo o tempo) e para a inclusão, como objectos *media*, de animações pré-definidas de elementos da cena.

Está também prevista a análise do desempenho e da funcionalidade de núcleos alternativos ao *RenderWare*, em particular o *Direct3D* [Tho96] da *Microsoft* e o *Heidi* da *AutoDesk*.

Um dos objectivos seguintes consiste na elaboração de um *plug-in* específico para o *3D Studio MAX* onde exista uma maior interacção nas especificações de atributos específicos para o *GAVE*, tentando reduzir ao mínimo o preenchimento manual do ficheiro de configuração, isto é, edição de texto.

O controlo completo da aplicação de RV em termos de toda a interacção existente, através de uma linguagem declarativa não procedimental é outro objectivo, de índole mais geral, que está planeado. A base desta linguagem será uma especificação poderosa da inter relação

entre eventos.

Este último objectivo encerra, em si, um perigo latente a evitar, que consiste na “sobre-especificação” em que a complexidade da especificação é tão grande que torna todo o *GAVE* pouco prático, vindo contra a sua directiva principal e fundamental.

## 5. Conclusão

Neste artigo foi apresentado o *GAVE - Generic Abstraction for Virtual Environments*, uma aplicação que permite a montagem automática de um sistema de *hiper-realidade* virtual. O potencial de acessibilidade desta metodologia a um leigo prende-se com o facto de se reutilizar a *interface* de um programa de modelação bastante divulgado para parametrizar todo o ambiente que se pretende visualizar. Na fase de parametrização são definidos elementos chave de um ambiente sintético como a configuração de luzes ou a especificação de segmentos de colisão.

O complemento ao trabalho de modelação e parametrização é a descrição das diversas propriedades do sistema num ficheiro de configuração com uma sintaxe pré-definida. No ficheiro de configuração é possível definir as propriedades de uma vasta gama de atributos do sistema que vão desde a configuração dos dispositivos de entrada e saída do sistema até à especificação dos eventos multimédia ou alteração das características de objectos.

Nas secções finais do artigo descreveu-se a utilização do *GAVE*, pelos autores, na construção de duas aplicações de RV que estiveram expostas ao público: *Salas da História* e *KM*. A elaboração destes dois projectos permitiu-nos confirmar que a redução em custos de integração que o uso do *GAVE* proporciona permite uma realização mais rápida e redirecciona o foco do desenvolvimento para o próprio conceito e *design* do ambiente virtual.

Devido às peculiaridades do *GAVE* em relação a outros pacotes e *toolkits* de RV cremos que o *GAVE*, tal como foi exposto, é uma ajuda para profissionais que embora não possuam grande formação informática têm o desejo ou a necessidade de utilizar Realidade Virtual nas suas áreas de actividade, tanto ao nível promocional como simulacional.

## Apêndice

Exemplo de ficheiro de configuração de uma aplicação:

```
#
# Ficheiro de Configuração de Propriedades
# Directórios
#
TEXTURE_PATH = ..\Objects
SCRIPT_PATH = ..\Objects
# Som
#
3DSND_DEV = 2
# Setup do DirectSound
```

```

#
DIRECTSOUND_OBJ = 0 ../Objects/MV01.WAV 0 1
DIRECTSOUND_FORMAT = 16 22050
COLLISION_SOUND_START = SOURCE_PLAY 0,12,0,0,0,0,0,0,0
COLLISION_SOUND_STOP = SOURCE_STOP 0
# Janela
#
WINDOW_SIZE = 640 480
# Câmara
#
CAMERA_NEAR_CLIPPING = -0.05
CAMERA_PAN = 270.0
CAMERA_TILT = 0.0
CAMERA_BACK_COLOR = 0.0 0.0 0.0 # R G B
CAMERA_VIEW_WINDOW = 0.9975 0.75 # width height
CAMERA_POSITION = -15.0 1.8 -7.5
# Estéreo
#
EYE_SEPARATION = 0.0375 ; 6 cm
FOCAL_DISTANCE = 1.375 ; 2.20 meters
# Temporizador
#
TIMER = 1
# Teclas
#
KEY_SENSIBILITY = 0.1 5
# Joystick
#
MOUSE_AS_JOYSTICK = 0.4
JOYSTICK_SENSIBILITY = 0.4 40
JOYSTICK_THRESHOLD = 5000 60000
# Cenas
#
SCENE = SALAS salas.ini 10 0 -1 0 0 0
SCENE = AUDITORIO auditorio.ini 10 0 -1 0 0 0
START_SCENE = SALAS

```

Exemplo de ficheiro de configuração de uma das cenas (salas.ini):

```

#
# Cena para SALAS
# Objectos estáticos
#
CLUMP = ../objects/salas.rwx
# Opacidade
#
FOREIGN_OPACITYCLUMP = ../objects/salas.rwx espelhoagu 0.5
# Posição Inicial
#
FOREIGN_INITIAL_POSITION = ../objects/salas.rwx InitPos
# Objectos media
#
FOREIGN_MEDIACLUMP = photo04 photo04 7 -1 10 -1 -1 None
FOREIGN_MEDIACLUMP = Sound01 Sound01 0 -1 0 -1 -1 None
# Segmentos para colisão
#
FOREIGN_SEGMENT = lim1 lim2
FOREIGN_SEGMENT = lim2 lim3
FOREIGN_SEGMENT = lim3 lim1
# Hotspots para objectos media
#
FOREIGN_POLYGON = photo04 4: Ph11Hot01, Ph11Hot02, Ph11Hot03, Ph11Hot04
FOREIGN_POLYGON = Sound01 4: Snd01Hot01, Snd01Hot02, Snd01Hot03, Snd01Hot04
# Portais para mudar de cena
#
FOREIGN_PORTAL = AUDITORIO 4: Pr01Hot01, Pr01Hot02, Pr01Hot03, Pr01Hot04
# Luzes
#
FOREIGN_LIGHT_CONICAL = luz11 alvoluz11 90 1 1 1 0.2
# Sons media
#

```

```
MEDIA_SOUND = Sound01 FIN_START 0 0 SOURCE_PLAY 1,1,0,0,0,0,0,0,0
MEDIA_SOUND = Sound01 FOUT_START 0 0 SOURCE_STOP 1
MEDIA_SOUND = Sound01 FOUT_END 0 0 SOURCE_STOP 1
```

## Bibliografia

- [Ell96] Elliott, Steven e Phillip Miller, "Inside 3D STUDIO MAX", New Riders, 1996.
- [Fer95] Ferraro, Richard F., "Learn 3D Graphics Programming on the PC", Addison-Wesley, 1995.
- [Fol92] Foley et al, "Computer Graphics", Addison-Wesley, 2nd edition, 1992.
- [Kal94] Kalawski, Roy S., "The Science of Virtual Reality and Virtual Environments", Addison-Wesley, 1994.
- [Ndime] Ndimension Home Page, <http://www.ndimension.com>
- [Nie94] Nielsen, Jakob, "Multimedia and Hypertext", Academic Press 1994.
- [Perfo] Performer Home Page, <http://www.sgi.com/Technology/Performer/>
- [Pro97] Proença, José A., "Áudio para Realidade Virtual", Trabalho Final de Curso, Licenciatura em Matemática Aplicada e Computação, IST, 1997.
- [Sense] Sense 8 Home Page, <http://www.sense8.com>
- [Theme] ThemeKit Home Page, <http://www.themekit.com>
- [Tho96] Thompson, Nigel, "3D Graphics Programming for Windows 95", Microsoft Press, 1996.