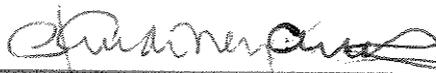


UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE ARTES

**O JOGO BLACK DEATH: UMA EXPERIÊNCIA DE DESIGN SONORO E
DESIGN DE JOGOS DE COMPUTADORES**

GUSTAVO QUEIROZ DE OLIVEIRA

*Este exemplar é a redação final da
dissertação defendida pelo Sr. Gustavo
Queiroz de Oliveira e aprovada pela
Comissão Julgadora em 20/08/2002*



Prof. Dr. Claudiney Rodrigues Carrasco
-orientador-

CAMPINAS – 2002

UNIDADE	BC
Nº CHAMADA	T/UNICAMP
	OL4j
V	EX
TOMBO BC/	51626
PROC.	16-837/02
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	04-12-02
Nº CPD	

CM00176893-8

BIB ID 271570

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IA - UNICAMP

OL4j

Oliveira, Gustavo Queiroz de.

O jogo Black Death: uma experiência de design sonoro e design de jogos de computadores / Gustavo Queiroz de Oliveira.
– Campinas, SP: [s.n.], 2002.

Orientador: Claudiney Rodrigues Carrasco.

Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Artes.

1. Multimídia. 2. Jogos eletrônicos. 3. Música. I. Carrasco, Claudiney Rodrigues. II. Universidade Estadual de Campinas. Instituto de Artes. III. Título.

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE ARTES
Mestrado em Multimeios

**O JOGO BLACK DEATH: UMA EXPERIÊNCIA DE DESIGN SONORO E
DESIGN DE JOGOS DE COMPUTADORES**

GUSTAVO QUEIROZ DE OLIVEIRA

Dissertação apresentada ao Curso de Mestrado em Multimeios do Instituto de Artes da UNICAMP como requisito parcial para a obtenção do grau de Mestre em Multimeios sob a orientação do Prof. Dr. Claudiney Rodrigues Carrasco.

CAMPINAS – 2002

À Luciane,
sem a qual este trabalho não passaria de um sonho.

"It's time for Rock`n Roll and chewing gum! And I've got a whole lotta gum!"

Duke Nuken em Duke Nuke Atomic Pack

"I see the games I develop as creating situations in which game players can utilize their own creativity to succeed. When the player's creativity can lead them to solutions, which I had not envisioned, it shows me that my game is doing its job." Richard Rouse III. Game

Designer em Gamasutra

"Maranaaahhh!!!!"

Devilish Priest em Blood,

Monolith Production

"Só se vive uma vez!"

Ally Mcbeal

"Sex ain't all I think 'bout. It's all I think 'bout You!"

Prince em The Gold Experience.

"The way I like it, the way it is

I got mine! Don't worry 'bout his!"

James Brown em Sex Machine

"A frase do Otto, Peixoto!! A frase do Otto, Peixoto!"

Nelson Rodrigues em "A Frase de Otto de Lara Resende
ou Bonitinha, mas ordinária"

"Tudo tem seu valor"

Anônimo

Agradecimentos

A todos aqueles apoiaram a realização deste projeto e em especial:

Ao Prof. Dr Claudiney Carrasco pela orientação, confiança e paciência.

Aos membros da banca pela atenção e gentileza despendidas e pelo apoio durante a graduação e o mestrado.

À CAPES pela bolsa de mestrado concedida.

À Nelson Pinton pela trilha musical maravilhosa.

Às Famílias Oliveira e Queiroz.

À Família Kawashima.

Aos amigos da grande república de *beta-tester* da PQP.

Aos amigos que sempre incentivaram e acreditaram neste projeto.

RESUMO

O Jogo Black Death é uma experiência de design de jogos de computadores que tem como objetivo demonstrar a importância da trilha sonora na articulação do ato de jogar.

Este trabalho tem um caráter híbrido, pois é composto de um jogo acompanhado por uma dissertação que apresenta as reflexões sobre o processo de criação.

A criação do jogo tem como referência os jogos comerciais de diversos gêneros no período de 1994 a 2002. O design do jogo Black Death foi planejado para que pudesse acomodar intervenções musicais nos moldes mais sofisticados de uso que a trilha sonora musical tem no cinema e no vídeo.

A dissertação apresenta um histórico dos jogos de computadores de 1958 ao momento presente. Posteriormente é apresentada uma descrição das diversas etapas da criação do jogo e por fim uma reflexão sobre o uso da trilha sonora em jogos comerciais e no jogo desenvolvido pelo pesquisador.

ABSTRACT

The game Black Death is an experience for computer game design based on Music Soundtrack biases. The way that Music and the Level Design interacts during the *gameplaying* is supposed to show how the Music has an important role on leading the player through the game.

The game is also presented with the Design Texts and the appointment of its production. The bases for the whole Design were the top selling computer games from 1994 to 2002. The game was conceived and designed for achieving the high standards that the Music Soundtrack has achieved on Movies.

The first chapter shows a brief historical review from beginning of electronic games in 1958 until nowadays. The second goes through the Design methods and Tools. And the third covers the Game Music Soundtrack specifically.

SUMÁRIO

Capítulo 1 - A História dos Jogos de Computadores	1
O Nascimento dos Jogos Eletrônicos	2
O Primeiro Vídeo Game	4
O Nascimento da Indústria de Jogos Eletrônicos	6
Surge a ATARI	7
Atari 2600	8
1980 Exodus na Atari	11
A Namco Lança Pac-Man	11
1982 – 1984 O Crash da Indústria de Videogames	11
Os Episódios Pac-Man e E.T	12
Os Jogos em Texto	13
Zork	14
On-line Systems, Sierra On-line, Sierra	14
A Interface Point-and-Click	17
Outros Aspectos da Evolução das Interfaces	19
O Nascimento dos Jogos de Ação em Primeira Pessoa – First Person Shooter	20
A indústria de Jogos Eletrônicos Hoje	23
Apresentação dos Jogos Analisados	24
Capítulo 2 - O Jogo Black Death	29
O Início do Projeto	30
Argumento	31
Como se Faz um Jogo Eletrônico	34
Uma Nova Estratégia	37
As “Engrenagens” de um Jogo de Computador	37
O Editor de Jogos	40
UnrealED – Apresentação e Características	44
Black Death	47
Criação de Ambientes Virtuais	50
Objetos 3D	52
Texturas	54
Luz	58
Áudio	63

Interação	65
Mover	65
Triggers	70
Arranjos e Redes de Triggers	71
Capítulo 3 – A Música nos Jogos de Computadores	77
Design Sonoro	78
Música para Jogos	78
Uma Classificação Possível	80
As Tecnologias de Áudio e Design Sonoro dos Jogos	82
A Trilha Sonora do Jogo Black Death	87
O ModPlug Tracker	88
Situação de Contorno	93
Onde inserir a música	94
Considerações Finais	100
Bibliografia	102

LISTAS DE FIGURAS E QUADROS

Figuras

Capítulo 1

Figura 1.1 - O Primeiro Jogo Eletrônico	2
Figura 1.2 - Spacewars	3
Figura 1.3 - O Primeiro Videogame	5
Figura 1.4 - Computer Space	6
Figura 1.5 - PONG	7
Figura 1.6 - Os Clones de PONG	8
Figura 1.7 - O ATARI 2600 e Alguns de Seus Jogos	10
Figura 1.8 - Da Interface Textual até Interface 3D	16
Figura 1.9 - Jogos com Interface Point-and-Click	18
Figura 1.10 - Pitfall	19

Capítulo 2

Figura 2.1 - Authorware - Exemplo de Orquestração e Produto Final	36
Figura 2.2 - Exemplo de Editores: Worldcraft e Q3Radiant	42
Figura 2.3 - Dois jogos, um Engine: Half-Life e Vampire - The Masquerade: Redemption	43
Figura 2.4 - UnrealEd: Biblioteca de Classes e o UnrealScript de um trigger	45
Figura 2.5 - UnrealEd - Visão da Interface	46
Figura 2.6 - A construção de um Arco Românico, passo-a-passo usando o Unreal Ed	53
Figura 2.7 - Aplicação de Texturas: mesmo objeto com texturas diferentes	55
Figura 2.8 - Texturas: O Vitral do Profeta	57
Figura 2.9 - Propriedades da Luz no UnrealEd	59
Figura 2.10 - Alguns Efeitos de Luz do UnrealEd	60
Figura 2.11 - Catacumbas: imagem só com texturas e imagem do jogo pronto	61
Figura 2.12 - A Luz do Vitral	62
Figura 2.13 - Mover	67
Figura 2.14 - As Propriedades de um Mover	69
Figura 2.15 - Propriedades de um Trigger	71
Figura 2.16 - Arranjos entre Triggers e Movers	73

Figura 2.17 - O Pilar Escavado	75
Figura 2.18 - Rede de Triggers	76

Capítulo 3

Figura 3.1 - Cripta Templária	95
Figura 3.2 - Catacumbas	96
Figura 3.3 - Corredores	97
Figura 3.4 - Prova Templária	98
Figura 3.5 - Torre	99

Quadros

Capítulo 1

Quadro 1.1 – Os Passos de Renderização em Wolfenstein 3D	22
----------------------------------------------------------	----

Capítulo 2

Quadro 2.1 – Mapa da Fase da Catedral	49
---------------------------------------	----

Quadro 2.2 – Organograma para Design de Ambientes Virtuais	51
------------------------------------------------------------	----

Capítulo 3

Quadro 3.1 – Interface do Modplug Tracker – Principal	90
-------------------------------------------------------	----

Quadro 3.2 – Interface do Modplug Tracker – Amostras de Áudio	91
---------------------------------------------------------------	----

Quadro 3.1 – Interface do Modplug Tracker – Edição de Patterns	92
----------------------------------------------------------------	----

INTRODUÇÃO

O futuro dos jogos eletrônicos é promissor, pois se verifica sua crescente importância e presença em nossa sociedade - modificando a forma das pessoas viverem, trabalharem e se divertirem.

Quem imaginaria que depois de pouco mais de trinta anos de seu surgimento, os jogos eletrônicos estariam tão presentes em nosso dia a dia. Atualmente, percebe-se a influência dos jogos em filmes e livros. Ao mesmo tempo esses mesmos filmes e livros servem de referencial para esses jogos. Seu conteúdo não têm distinção de credo, etnia e tempo, se apropriando de vários elementos da cultura contemporânea, como a música dos templos e os grafites das metrópoles.

A indústria tem respondido a esta demanda com tecnologias que possibilitam a criação de jogos mais interativos, com níveis de imersão e desafio cada vez mais atraentes ao público. Os personagens e os "mundos" dos jogos tornam-se cada dia mais realistas, aumentando a ilusão de se estar interagindo com pessoas e ambientes "reais". Provavelmente, a música e o som não serão simplesmente barulhos de fundo. Serão elementos essenciais para criar emoção e empatia, reproduzindo sensações da vida real ao mesmo tempo em que conquistam a imaginação do jogador.

Fazer um jogo é um grande desafio. O designer de jogos tem de dominar o aspecto prático de seu ofício. A integração de conhecimentos de várias áreas como: arquitetura, computação e música traz por si só um volume considerável de trabalho. Não é à toa que qualquer jogo comercial conta com uma equipe de desenvolvimento, com dezenas de profissionais trabalhando direta e indiretamente no projeto por vários meses, ou até anos.

O presente trabalho foi a oportunidade de entrada nesse universo. E a prática diária nos obrigava a aprimorar conceitos de design, de tecnologia, de metodologia e estética durante toda a produção do jogo.

Para adequar este volume de trabalho ao período do mestrado, decidimos realizar somente a fase da Catedral de Gênova, pois julgamos que ela seria suficiente para experimentar grande parte de nossas idéias e hipóteses.

O produto deste trabalho é um CD-ROM com a fase da Cathedral para ser jogada sobre a plataforma do jogo Unreal da Epic Games, acompanhada do texto da Dissertação de Mestrado.

A dissertação está estruturada em três capítulos. O Capítulo 1, traz uma breve história dos jogos eletrônicos desde o seu surgimento até a criação do jogo de ação em primeira pessoa (*first person shooter*). O objetivo é mostrar que o jogo de computador herda elementos da indústria de videogames com elementos dos jogos em textos para vários jogadores desenvolvidos nos computadores das universidades americanas a partir da década de setenta. O Capítulo 2 descreve processo de design do jogo Black Death desde a concepção de seu roteiro, abordando pontos de execução de edição de imagens, modelagem de ambientes tridimensionais até tópicos específicos de softwares para criação e programação de jogos de computadores. O Capítulo 3 contém a análise da trilha sonora dos jogos comerciais e dados técnicos de formatos de áudio para jogos. Apresenta uma descrição de como foi planejada e executada a trilha sonora para o jogo Black Death e a análise dos resultados alcançados. Cada etapa do processo de criação traz além dos conceitos, ilustrações com exemplos práticos e genéricos, que podem ser usados em qualquer situação, independente de programas e plataformas.

A intenção deste trabalho é apresentar conceitos básicos para aqueles que se interessem em iniciar uma pesquisa, um trabalho ou apenas aprofundar o conhecimento nessa área. Esperamos que esta contribuição possa multiplicar o interesse pelo estudo dos jogos eletrônicos no meio acadêmico, desmistificando a imagem de uma arte inacessível, que depende de conhecimentos e tecnologias além da capacidade de nossas universidades.

Capítulo 1 – A História dos Jogos de Computadores

O Nascimento dos Jogos Eletrônicos

Fumante inveterado, jogador confesso de *pinball*, ele só queria criar algo para divertir as pessoas que visitavam o *Brookhaven National Laboratories*. Assim, William Higinbotham criou em 1958 o primeiro jogo eletrônico: *“Tennis for Two”*. A visitação pública a centros de pesquisa, em dias especiais, era uma prática comum naquele tempo de Guerra Fria. Os centros de pesquisa precisavam mostrar ao povo o poderio tecnológico de seu país, justificando assim os seus gastos.

“Tennis for Two” usava um osciloscópio como “visor” e era processado por um computador analógico. Seu inventor nem chegou a patentear-lo, pois não vislumbrou nele nenhum potencial comercial. Willy dizia que qualquer um poderia ter feito aquilo. Morreu em 1995 sem o devido reconhecimento.

Por muito tempo, o jogo considerado como o primeiro de todos foi: *“Spacewars”*. Desenvolvido em 1961 por J. Martin Graetz, Stephen Russell e Wayne Witanen do MIT - Massachusetts Institute of Technology. Criado para mostrar toda a capacidade tecnológica que era o DEC PDP-1, o primeiro minicomputador fabricado, o jogo utilizava conceitos de Física como Aceleração e Gravidade para recriar uma batalha espacial. Sucesso absoluto de visitação. *Spacewars*, empolgou tanto a DEC - Digital Equipment Corporation que todo novo PDP-1 vendido era testado com *Spacewars* antes de chegar ao seu destino. Devido ao tipo de memória que o PDP-1 usava, se nenhum problema ocorresse no transporte, o computador poderia ser ligado e rodar *Spacewars*! Todos os programas feitos pelo alunos do MIT no PDP eram de domínio público e as fitas de papel com os códigos seguiam junto com cada novo equipamento.

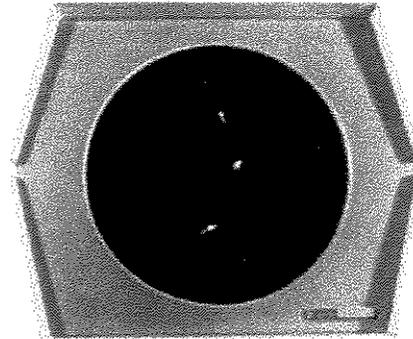


Figura 1.1 -
O primeiro Jogo
Eletrônico

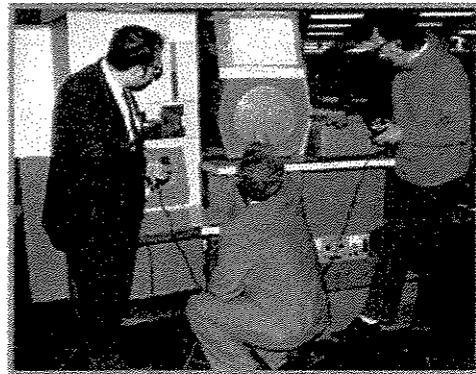
Spacewars



(a) O PDP-1



(b) Spacewars



(c) J.M. Graetz, Alan Kotok e Steve Russell



(d) O livro de E.E. "Doc" Smith " The Skylark of Space" que inspirou Spacewars

Figura 1.2 - Spacewars -

(a) O Digital Equipment Corporation PDP-1 ocupava uma sala inteira no MIT.

(b) Monitor com a imagem do jogo pronto.

(c) Os criadores de Spacewars em um encontro nos anos 70, da esq. para dir. J. Martin Graetz, Alan Kotok, responsável por "conseguir" a função Seno-Cosseno que controlava os movimentos das espaçonaves e Steve Russell, pai do projeto e programador.

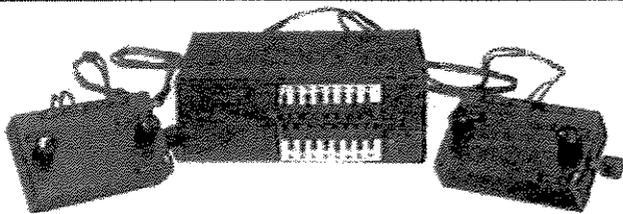
(d) O livro que inspirou Spacewars.

O Primeiro Vídeo Game

A primeira patente de um videogame só foi feita em 1968 pelo físico alemão radicado nos Estados Unidos Ralf Baer, sua intenção era criar um dispositivo barato que, através de um televisor, rodasse jogos eletrônicos. O nome do protótipo era "Brown Box".

O console foi conhecido mais tarde como Odyssey e era fabricado pela Magnavox, empresa pertencente ao grupo holandês Phillips. Uma curiosidade desse aparelho é que devido a suas limitações ele não conseguia gerar todos os pontos necessários para preencher a tela do televisor e para remediar esta "deficiência" o usuário precisava encaixar cartões plásticos na tela de sua Tv.

O Primeiro Videogame



(a) O protótipo do BrownBox - Sanders Associates, 1969



(c) O Odyssey 100



(b) Ralph Bauer

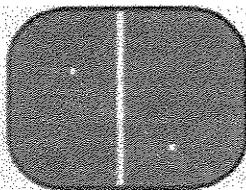
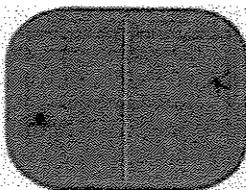
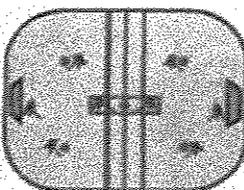


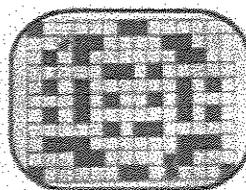
TABLE TENNIS. The basic Odyssey game that develops your electronic coordination.



TENNIS. All the excitement of Wimbledon as you serve, volley and score.



HOCKEY. Face-off, dig for the net, maneuver the puck goal!



CAT AND MOUSE. Can the clever mouse evade the cat-bentious cat? A hilarious electronic addition game of wits and wack.

(d) Os cartões para colocar no televisor.

Figura 1.3 - O Primeiro Videogame

(a) O protótipo do BrownBox apresentado por Ralph Bauer à diretoria da Sander's Associates em 1969.

(b) Ralph Bauer ao lado do caixa do Odyssey 100 em 1972.

(c) O Odyssey 100 em sua versão comercial, custava u\$100,00.

(d) Os cartões de plástico deviam ser colocados no televisor para completar o visual dos os jogos.

Os jogos do Odyssey 100 não tinham som.

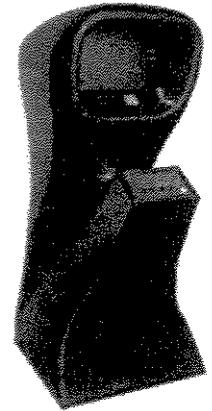
O Nascimento da Indústria de Jogos Eletrônicos

Nenhuma pessoa foi mais importante para o nascimento da indústria de jogos eletrônicos que um jovem estudante de engenharia que passava a maior parte do tempo “jogando” *Spacewars* no PDP-1 na Universidade de Utah, uma das três únicas universidades americanas que tinham condições de possuir tal equipamento. **Nolan Bushnell**, perceberia as possibilidades comerciais de um jogo eletrônico, como *Spacewars*, depois de um emprego de verão em um Fliperama de um parque de diversões em Salt Lake City. Ao ingressar na universidade, iniciou um projeto pessoal que consumiria oito anos de sua vida: produzir uma versão arcade¹ de *Spacewars*. Chegou ao extremo de transformar o quarto de sua irmã mais nova em seu laboratório, só para poder trabalhar neste projeto. Depois de graduado, em 1970 é contratado pela AMPEX (empresa de áudio/vídeo criadora do videotape em 1957) e onde conheceu **Ted Dabney**. O companheiro de trabalho se empolga com a idéia e ajuda Bushnell nas horas de folga. Naquele mesmo ano, Bushnell se convence de estar no caminho certo e larga o emprego para trabalhar integralmente no projeto. Em 1971, eles conseguem criar o primeiro fliperama eletrônico: **Computer Space**. Logo encontram uma empresa disposta a fabricar a máquina, a **Nutting Associates**. Foram fabricadas mil e quinhentas máquinas de **Computer Space**, mas muito poucas foram vendidas, a Nutting Associates pressiona Bushnell e o envia a feira da **Magnavox Profit Caravan** em maio

de 1972 no Hotel Airport Marine em Burlingame na Califórnia, para verificar os rumores de um sistema doméstico para jogos eletrônicos. Lá ele conhece **Ralph Bauer** e joga **Tennis** no **Odyssey** por mais de meia hora. Bushnell volta convencido que qualquer jogo para fazer sucesso deve ser muito simples, pois o excesso de comandos e controles do



Nolan Bushnell



Computer Space - Nutting Associates, 1971.

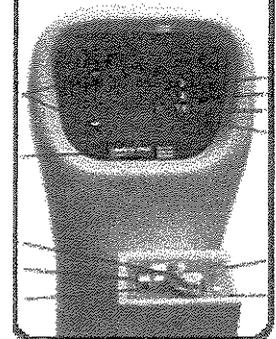


Figura 1.4 - Computer Space

¹ O termo Arcade denomina as máquinas de jogos feitas para casas comerciais, os fliperamas.

Computer Space eram muito complexos para os fregueses semi embriagados dos bares da América.

Surge a ATARI

O ano da revolução dos jogos eletrônicos foi 1972. Sem apoio na Nutting Associates, Nolan Bushnell e Ted Dabney fundaram a ATARI com o investimento inicial de U\$ 250,00 para cada sócio. O nome é inspirado no jogo de tabuleiro japonês "GO" e tem o mesmo significado que xeque-mate no Xadrez. A objetivo da empresa era o de criar e distribuir jogos (software) para as outras companhias, mas o mercado de Arcades tradicionais resiste à idéia e Bushnell decide comercializar ele mesmo suas máquinas com seus programas.

A Atari produz **PONG**², um jogo do tipo tênis de mesa, onde cada jogador comandava uma barra vertical que fazia o papel de "raquete". A simplicidade do jogo cria uma verdadeira febre de costa a costa, despertando o gosto e curiosidade do público para as diversões eletrônicas. Em um ano, mais de oito mil e quinhentas máquinas são vendidas, muito mais do que a média de um arcade de sucesso da época que era 2000 unidades.

Logo inicia-se um movimento que seria comum na indústria: a cada lançamento surgem vários clones de cada sucesso. A concorrência cresce e a Atari precisa se mexer.

A seguir, Buchnell dá um passo genial, lança o **Home PONG**, a versão do jogo do momento para aparelhos domésticos que podia ser usado em qualquer televisor. Com a parceria da gigante loja de departamentos americana SEAR'S, a Atari cresce e torna viável o mercado de jogos eletrônicos. Naquele ano foram vendidos mais de

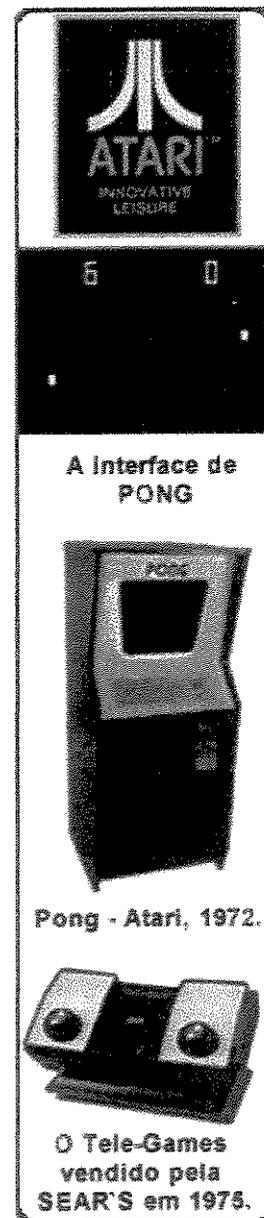


Figura 1.5 - PONG

² O nascimento de Pong - Bushnell contrata Al Alcorn, um programador inexperiente e como exercício pede que ela faça um jogo estilo video tennis. Referências não faltam, como "Tennis programmig" de Willy Higginbotham e "Tennis" do Odyssey 100. Depois de pronto, ambos concordam em chamá-lo de Pong por dois motivos: primeiro o nome Ping-Pong já era uma marca registrada, segundo "pong" era o único som que o jogo fazia.

150.000 consoles do Atari, e seguindo este movimento surgiram vários clones do Home Pong (fig.1.6).



Figura 1.6 - Os Clones de PONG

Atari 2600

Em 1975, a Atari possuía um faturamento de 40 milhões de dólares anuais, atraindo a atenção do gigantesco conglomerado de media Warner Communications. Naquele ano a Atari lançaria o seu primeiro console programável, o VCS (Video Computer System), mais tarde conhecido como 2600.

Em 1976, a Warner Communications comprou a Atari por U\$ 26.0000,00 com Bushnel ocupando o cargo de CEO (Chief of Executive Office) – Chefe Geral de Executivos (livre tradução). As vendas do Atari 2600 não decolaram no início, por dois anos o console não encontraria seu nicho de mercado fazendo com que as ações da Warner caíssem

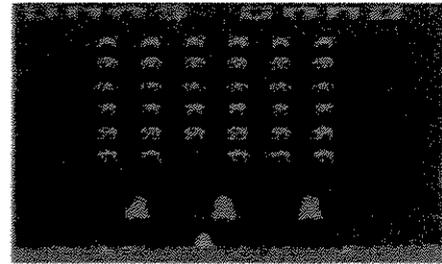
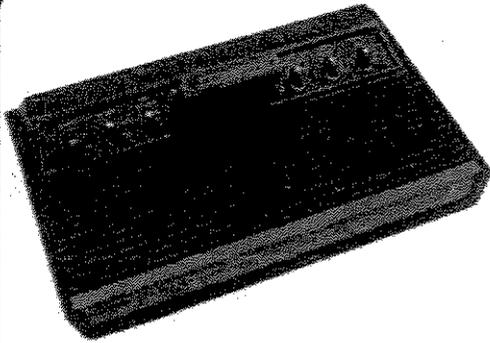
perigosamente. Depois vários atritos, a Warner despede Bushnell. A companhia alega que sua administração era muito relaxada e com baixa produtividade, somente doze títulos novos foram produzidos neste período.

O Grande salto da indústria de videogames foi dado pela Atari em 1980. Ela foi a primeira empresa a licenciar um jogo arcade para sistemas domésticos: Space Invaders da japonesa Taito. O público correu para as lojas em busca de Space Invaders, chegaram a comprar o console somente por causa daquele jogo. A Atari produziu ainda mais 112 variações do título.

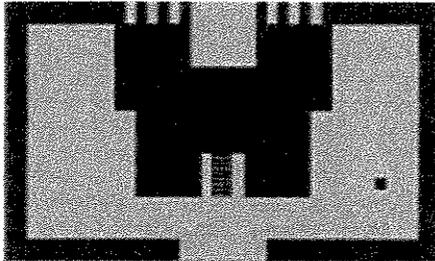
Inicia-se assim o domínio da Atari no mercado de videogames, mais de 25 milhões de unidades do 2600 foram fabricadas, rendendo mais de 5 bilhões de dólares para a empresa. A única rival significativa era a Mattel com o Intellivision. Os jogos eletrônicos (arcades e sistemas doméstico) estavam agora na linha de frente da indústria do entretenimento, movimentando cinco vezes o montante da indústria cinematográfica.

Este movimento quebrou a sazonalidade de vendas de videogames que acompanhava o mesmo ritmo da indústria de brinquedos. Os consoles eram produzidos o ano inteiro. Mais de 200 títulos novos eram lançados por ano por cerca de 40 fabricantes. Aproximadamente 120 milhões de cartuchos foram vendidos para 55 consoles compatíveis, com alguns títulos conseguindo projeção mundial. A Atari que no início foi considerada um desastre de investimento da Warner, hoje era responsável por 50% dos lucros deste conglomerado de comunicações e entretenimento.

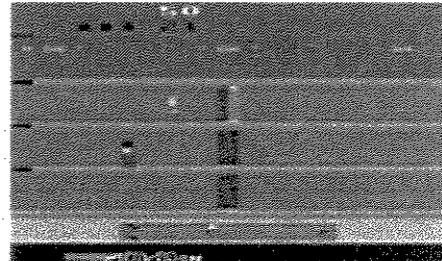
O ATARI 2600 e Alguns de Seus Jogos.



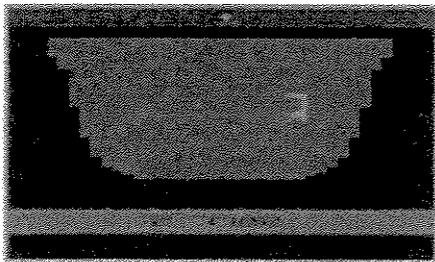
Space Invaders



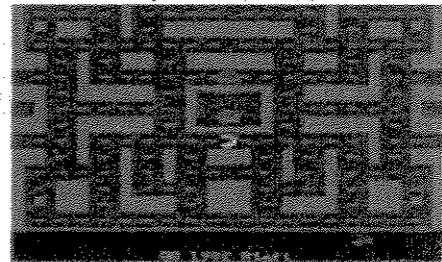
Adventure



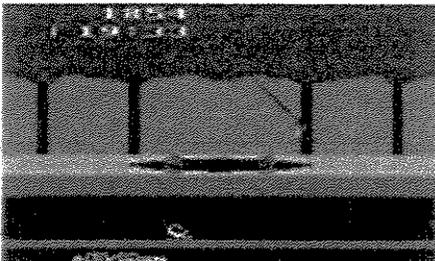
Keystone Keeper



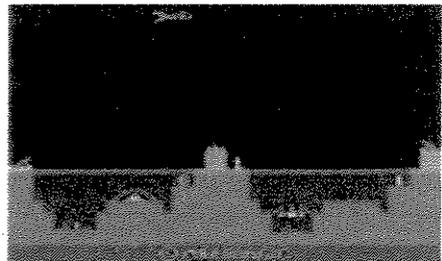
E.T.



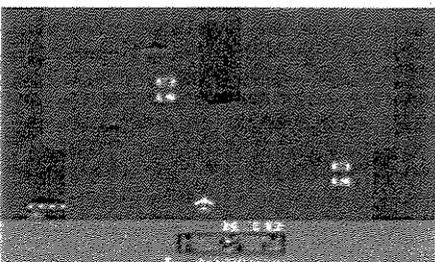
Pac-Man



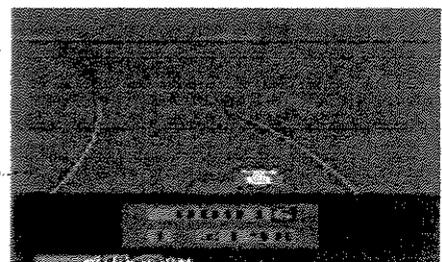
Pitfall



Atlantis



River Raid



Enduro

Figura 1.7 - O ATARI 2600 e Alguns de Seus Jogos.

1980 Exodus na Atari

A política da companhia de não incluir créditos de produção em seus cartuchos fez com que vários designers e programadores abandonassem a Atari e fundassem a Activision tornando-se a primeira empresa terceirizada de jogos eletrônicos. A Activision não só reconhecia o valor individual de cada desenvolvedor, como incluía seus nomes nos cartuchos e material de marketing. Lançando jogos de peso como: Pitfall, H.E.R.O, Enduro. Logo depois, mais profissionais da Atari e da Intellivision fundaram a Imagic responsável por títulos como: Atlantis, DemonAttack. Ao contrário da Activision, a Imagic durou apenas poucos anos e não conseguiu criar nenhum outro título que a sustentasse no mercado.

A Namco Lança Pac-Man

A empresa japonesa é responsável pelo arcade mais popular de todos os tempos. Mais de 3000.000 unidades são vendidas em todo mundo. Pac-Man é primeiro jogo a gerar uma febre de consumo para produtos de merchandising camisas, tolhas, xícaras, cereais, desenho animado (*Hanna-Barbera* 1982-1984). O fenômeno *Pacmania* se enraíza na cultura popular americana a ponto de ser capa da revistas *Times* e *Mad*.

1982 – 1984 O Crash da Indústria de Videogames

O grande *crash* da indústria de videogames (1982-1984) foi o período em que todas as grande companhias americanas de consoles quebraram ou foram vendidas. Neste época existiam 3 grandes fabricantes no mercado: a Colleco (Collecovision), Mattel (Intellivision) e Atari (2600). A base instalada era enorme, vinte e cinco por cento dos lares americanos possuíam pelo menos um dos sistemas. A cada semana “novos” títulos eram lançados sem nenhum controle efetivo de qualidade. O mercado estava tão saturado de empresas de ramos não relacionados a jogos que até a o grupo Quaker da Indústria de Alimentos tinha uma divisão de jogos. Milhares de consoles eram fabricados e o mercado não absorvia mais o excedente. Nas lojas, os cartuchos chegavam a ser vendidos por um

dólar! Este quadro mina a fonte de rendimentos das empresas. Alguns sinais já evidenciavam que a indústria como um todo não ia bem.

Em 7 dezembro, a Atari anunciaria que as vendas do 2600 não atingiriam as expectativas e a ações da Warner Communications caem 32% em um único dia. A Colleco e a Atari tentam diversificar seus ramos de atuação: a primeira entra no mercado de microcomputadores com o Adam e a segunda no de telecomunicações. Ambos os projetos fracassam. A Colleco pede falência em 1985. A gigante de brinquedos Mattell resolve abandonar sua divisão de jogos antes que o queda nas vendas afete todo o seu conglomerado.

A Atari nunca mais recuperou o prestígio no mercado, seus consoles posteriores foram fracassos seguidos, hoje todo os direitos dos jogos pertencem à Hasbro Interactive.

Os Episódios Pac-Man e E.T

A Atari lança então, a tão esperada versão de Pac-Man. Nem de longe o cartucho lembra o arcade, o público começa desconfiar da empresa. O próximo desastre foi o lançamento de E.T. para o 2600. Steve Ross, presidente da Warner negocia um contrato de 21 milhões de dólares com a Amblin Entertainment de Steven Spielberg, esperando um sucesso tão grande quanto o filme. A tarefa de criar o jogo mais caro de todos os tempos coube a Howard Scott Warshaw com o tempo exíguo de seis semanas.

Mais um fracasso absoluto. Chegaram às lojas 5 milhões de cartuchos, somente 1 milhão foram vendidos. Para agravar mais a situação, foram fabricados mais consoles. Caminhões cheios de cartuchos e console encaalhados tiveram de ser jogados em um aterro da empresa no Novo México, soterrando a hegemonia da Atari.

Por muito tempo, nenhuma empresa americana se arriscaria a chegar perto de um joystick. Mas em Kioto no Japão, uma empresa fabricante de cartas de baralho, fundada em 1889 tinha planos diferentes, seu nome: Nintendo.

Os Jogos em Texto

Mais uma vez no MIT, surgiria um conceito que mudaria o cenário do mundo dos computadores. O legendário prof. John McCarthy (pioneiro da Inteligência Artificial e criador da linguagem LISP), enfrentava o seguinte cenário em 1957:

- cada vez mais, seus projetos e alunos precisavam usar os computadores (só haviam dois disponíveis)
- os computadores continuavam muito caros e a compra de novos equipamentos não resolveria o problema.

Sua solução foi simples: dividir o tempo de uso do computador central (*mainframe*). Nascia a *Timesharing Computing*³. O computador dividia o tempo de acesso de cada terminal possibilitando que todos tivessem a impressão de estarem usando-o sozinhos. Isto implica que a capacidade do computador deveria ser dividida entre todos usuários. Aplicativos como *Spacewars* que consumiam integralmente a capacidade de processamento e memória da máquina não podiam ser executados em *Timesharing*. A interface torna-se textual, democratizando o uso e aumentando a produtividade por computador.

Durante as décadas de 60 e 70, os computadores com ambientes Timesharing se popularizaram pelas universidades americanas e começaram a surgir jogos textuais com grande influência das histórias de Ficção Científica, do RPG “Dungeons and Dragons”⁴ e da obra de J.R.R. Tolkien⁵.

Alguns jogos famosos desta época:

- “*Star Trek*” de Mike Mayfield na University of California - Irvine, 1971.

³ Computação de Tempo Compartilhado – livre tradução

⁴ Dungeons and Drangons – (Labirintos e Dragões), Gary Gygax e Dave Arson. D&D é sem dúvidas um dos RPGs de maior influência e sucesso no mundos dos jogos. Muito popular no inicio dos anos 70 nos EUA. Sua primeira publicação como livro de estratégias de RPG foi em, 1974.

⁵ A obra de J.R.R.Tolkien teve desdobramentos em muitos outros campos como: nas letras da banda inglesa Led Zeppelin, nos RPGs. Dungenos and Dragons é apenas um dos vários títulos que se inspiram nos personagens e cenários imaginados por Tolkien. Hoje presenciamos um retorno desta obra aos *mainstream* com a tão aclamada e esperada versão para o cinema da trilogia o “Senhor dos Anéis”. O primeiro livro que introduz este universo é “O Hobbit” em 1937. A trilogia o Senhor dos Anéis tem sua primeira parte “*The Fellowship of the Ring*”, publicada em 1954.

- “*Hunt the Wumpus*” de Gregory Yob na University of Massachusetts, em 1972.
- “*Adventure*” de Don Woods na Stanford University, 1972.

Esses jogos on-line vão povoar a ARPAnet, a famosa rede de computadores criada pelo governo americano como estratégia para preservar a governabilidade do país, caso houvesse um ataque nuclear russo. A ARPAnet conectava o governo e as universidades americanas, deu origem à atual Internet.

Zork

Zork foi criado pelos estudantes do MIT, Dave Lebling e Marc Blank, o jogo em texto evoca o universo de “*Dungeons and Dragons*” e rodava no então no *mainframe* PDP-10.

Com o surgimento dos pc e sua popularização a Dave e Marc começam a trabalhar numa forma de converter o jogo em um produto vendável que pudesse ser executado com os poucos recursos. Fundam a INFOCOM e em 1980 produzem a primeira versão de Zork. O contrato de distribuição foi feito com então renomada *software house* Visicorp, fabricante do primeiro programa de planilha eletrônica. São vendidas 1 milhão de cópias.

Com o aumento de capacidade dos pc`s e seu barateamento, surgem as interface gráficas, as pequenas empresas desenvolvedoras de programas para um mercado que não parava de crescer. O período é de franco desenvolvimento para as empresas de software, são desta época a Borland criadora do Delphi e a Adobe que se destacaria como uma das impulsionadoras da editoração eletrônica no mundo.

On-line Systems, Sierra On-line, Sierra

Em 1980, uma empresa formada pelo casal Ken e Roberta Williams, cria o primeiro jogo a combinar interface gráfica e textos para o Apple II. *Mystery House* era um jogo de mistério no estilo Agatha Christie, o jogador deveria vasculhar a casa em busca de tesouros enquanto tentaria evitar o destino fatídico dos moradores. Os gráficos eram linhas grosseiras e o programa que executava toda parte textual era a Z-machine da

Infocom. O jogo venderia 11.000 cópias. A empresa produziu mais de vinte jogos para o Apple II, chegando a receber uma carta elogiosa de Steve Wozniak, dono da Apple. Na carta ele fala dos benefícios dos jogos nas vendas de seus computadores.

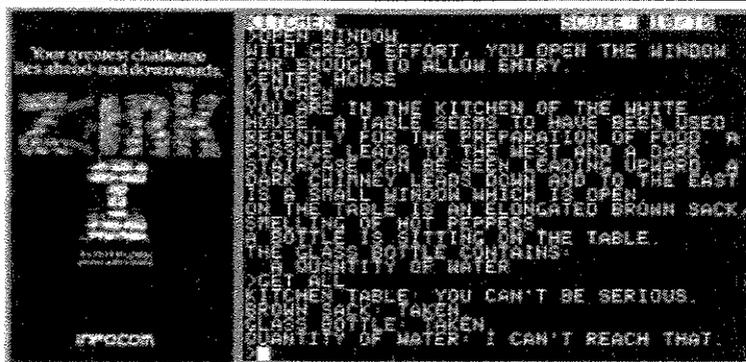
Em 1983, a IBM lança o PCjr., um microcomputador com vistas no mercado de jogos. Usando um protótipo, Roberta Williams desenvolve um jogo que permitia ao jogador explorar ambientes Pseudo 3D⁶ em incríveis 16 cores (para a época). Com uma equipe de seis programadores e um orçamento de US\$ 700.000, King's Quest é lançado em 1984. O profissionalismo dos desenvolvedores de jogos para computadores se concretiza, as empresas crescem e deixam de funcionar como pequenas empresas familiares. Aos poucos formam-se equipes de desenvolvedores organizados em vários setores como: design de fases (*level design*), arte, design sonoro.

CD-ROM, 256-VGA, Soundblaster

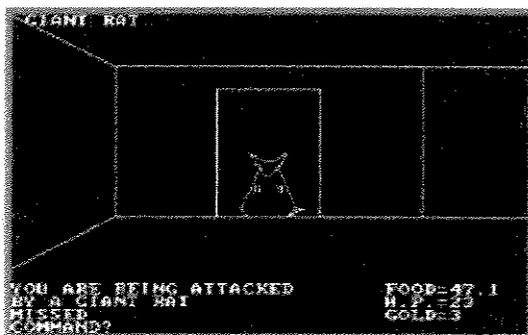
Foi em 1989 que a tecnologia que suporta os jogos até hoje se consolidou. As placas de vídeo de 256 cores e as placas de som Soundblaster e Adlib, tornaram-se cada vez mais populares. O CD-ROM recebe seu primeiro título - "*The Manhole*" da Activision.

⁶ O Pseudo 3D é na verdade um ambiente desenhado com perspectiva traçada a partir de um ponto fixo. O jogador se movimenta somente nos eixos X e Y. O personagens podem ser desenhados com escalas diferentes simulando profundidade.

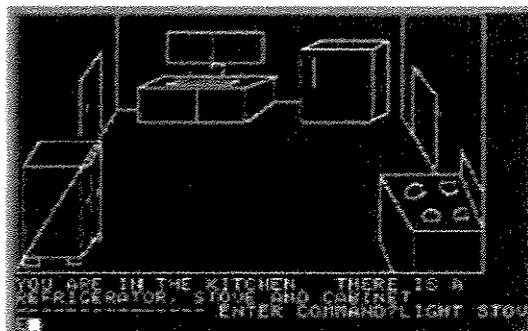
Da Interface Textual até Interface 3D



Zork - Infocom, 1980



Akalabeth - California Pacific, 1979.



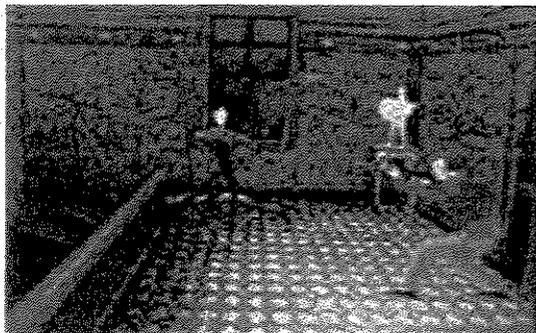
Mystery House - On-Line Systems, 1980.



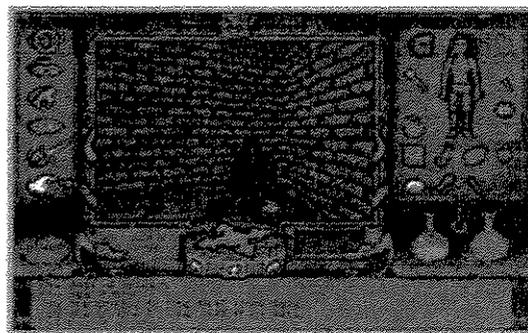
King's Quest - Sierra On-Line, 1984.



Maniac Mansion - LucasArts, 1987.



Alone in the Dark - Infogames, 1993.



Ultima Underworld - Origin, 1993.

Figura 1.8 - Da Interface Textual até Interface 3D

A Interface *Point-and-Click*

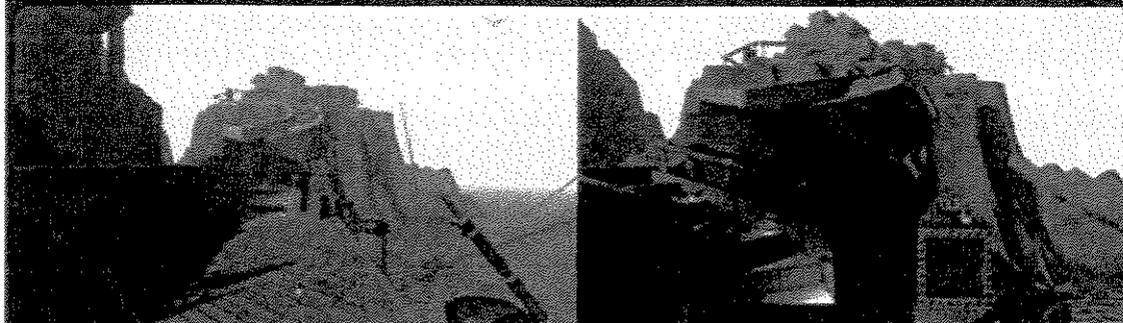
A interface conhecida como *point-and-click* (apontar-e-clicar) foi criada para jogos pela LucasArts em 1991 e muitos títulos de peso foram produzidos baseados neste tipo de interação: *The Dig* (LucasArts, 1993), a série *Gabriel Knight* (Sierra, 1995 a 1999), *Monkey Island* (LucasArts, 1994), a série *Wing Commander* (Origin, 1995), *Seventh Guest* (Trilobyte, 1993) entre outros.

O marco desta época foi *Myst*, o jogo em CD-ROM mais vendido em todos os tempos, responsável pela alavancagem dos mercado de CD-ROM. Nesta época era comum que os jogos tivessem vários CDs, (*WingComander III* possuía cinco CDs, *Gabriel Knight 2* possuía seis CDs). Este fato mostrava o quanto público desejava experimentar um grande volume de conteúdo e quanto estava disposto a passar mais de semanas jogando.

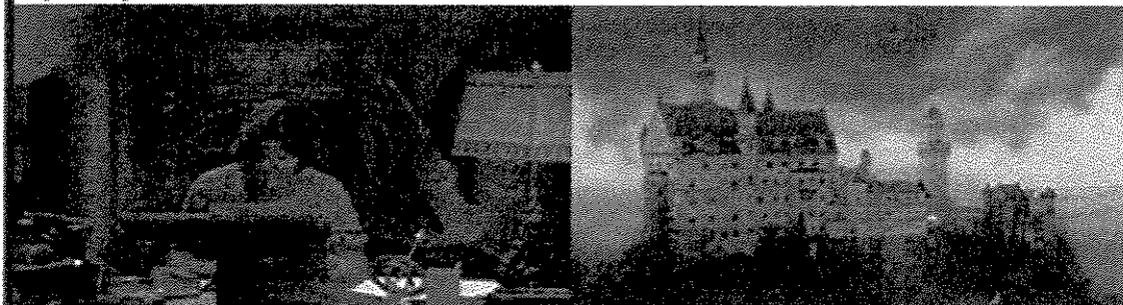
Foram estes jogos que consolidaram o mercado dos jogos de computadores e que fixaram muitos de seus padrões como: uso de filmes para contar parte do enredo (cinematics, ou cutscenes), uso de quebra cabeças e charadas, uso da seção de inventário (tela especial onde o jogador guarda todos os objetos coletados durante o jogo para usá-los posteriormente), uso combinado de objetos do inventário.

Hoje, mesmo com tecnologia muito mais avançada presenciamos que muitos jogos recentes como *Vampire: The Masquerade – Redemptiom*, usam uma mescla de elementos de vários gêneros como: ação em primeira pessoa (*first person shooter*), RPG e *Point-and-Click*. Para criar novas formas de jogar, os designer de VTMR, sobrepueram a mecânica do *Point-and-Click* à liberdade de movimentos do jogo ação em primeira pessoa.

Jogos com interface *Point-and-Click*



Myst - Cyan, 1993.



Gabriel Knight - The Beast Within - Sierra, 1995



7th Guest - Trolobyte, 1993



Vampire The Masquerade - Redemption - Nilistic, 1999.

Figura 1.9 - Jogos com interface *Point-and-Click*

Outros Aspectos da Evolução das Interfaces

Se analisarmos “PitFall” (David Crane, 1982) da Activision, veremos que o jogo apresenta um progresso expressivo na forma de apresentar o enredo. Em sua interface já se percebe ou pelo menos se intui, um caminho para o desenvolvimento de uma história.

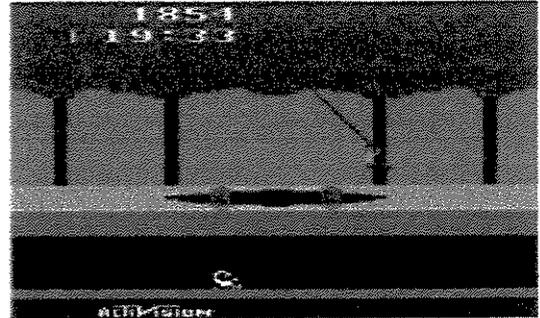


Figura 1.10 - Pitfall

O jogador está no papel de um explorador no estilo de *Indiana Jones* (1983) e tem vinte minutos para escapar da selva, mas em seu caminho estão troncos que rolam loucamente em sua direção, lagos repletos de crocodilos, precipícios a serem cruzados por meio de cipós e grutas subterrâneas tomadas de escorpiões.

O ambiente onde Pitfall se desenrola, não pode ser mostrado em apenas uma tela. Não se vê todo o ambiente onde o jogo ocorrerá. Em jogos como: *Space Invaders*, *Breakout* ou *Pac-Man*, a ação se desenrola sobre um único cenário, seja destruir os alienígenas invasores ou demolir o “paredão”, o enredo ou pelo menos o seu projeto, não tem dimensão de tempo, é pontual.

Pitfall apresenta um ambiente virtual onde é preciso “caminhar”, explorar, mesmo que apenas em duas direções: para frente ou para trás. Apesar desta diferença, o jogo continua com a característica dos obstáculos/inimigos que não param de vir na direção do jogador; mas apresenta múltiplos caminhos para serem completados. E, mais importante, possibilita a variação do ritmo da ação conforme o caminho escolhido.

Um descendente direto de Pitfall é “*The Prince of Persia*”. Por meio de animações ele apresenta um enredo em que um vizir maligno, seqüestra a princesa e aprisiona seu pretendente nas masmorras do palácio. Uma grande ampulheta conta o tempo para que a princesa se case com o vizir e seu amado seja morto.

O jogo apresenta uma interface em duas dimensões, mas que tem mostrar alguma profundidade de seus ambientes. Este efeito é dado por seu cenário desenhado em perspectiva e por meio de um “truque” muito inteligente, que consistia em fazer com que em algumas situações o personagem fosse mostrado atrás ou à frente do cenário.

Exemplo: O nosso príncipe caminha entre as colunas do palácio e é desenhado atrás das colunas mais próximas e na frente das colunas mais distantes, criando o efeito de profundidade.

O Nascimento dos Jogos de Ação em Primeira Pessoa – *First Person Shooter*

O Jogo de Ação em Primeira Pessoa (First Person Shooter) foi o gênero que mostrou pela primeira vez, em tempo real o jogo visto pela perspectiva do jogador, transportando-o para dentro do jogo, atingindo um nível de imersão totalmente novo para os jogos de computadores. O precursor foi *Wolfenstein 3D*.

Wolfenstein3D

Apesar de não ser exatamente o primeiro jogo 3D com a perspectiva de primeira pessoa, existiu *Battlezone* da Atari⁷, "*Wolfenstein 3D*"⁸ da Id Software inaugura a era dos jogos de ação em primeira pessoa e fixa padrões que serão seguidos e dados como certos do gênero até recentemente.

No jogo, você está na "pele" de um a soldado "Aliado" capturado na 2ª Guerra Mundial em um castelo Nazista. Seu objetivo é escapar com vida da prisão. Durante sua fuga ensangüentada, o jogador deve coletar tesouros, lutar contra soldados, cães e até o próprio Führer.

Na interface, as partes visíveis do jogador são: suas mãos, sua arma e seu rosto que fica localizado fora da área de ação, abaixo no painel de status. Existem áreas específicas para a condição de saúde (*Health Status*), quantidade munição e opções de armas.

O ambiente virtual onde ocorre a ação será chamado daqui por diante de "fase" do inglês *level*. Este ambiente contém partes móveis que podem ser acionadas pelo

⁷ *Battlezone* (Atari, 1980) era um simulador Arcade de Tanque de Guerra com gráficos vetoriais desenhados na perspectiva de primeira pessoa. Foi um grande sucesso e até o exército norte-americano encomendou unidades especiais do jogo para serem usados no treinamento de seus soldados.

⁸ *Wolfenstein 3D* é inspirado no jogo para Apple II "*Castle Wolfenstein*" de Silas Warner de 1982.

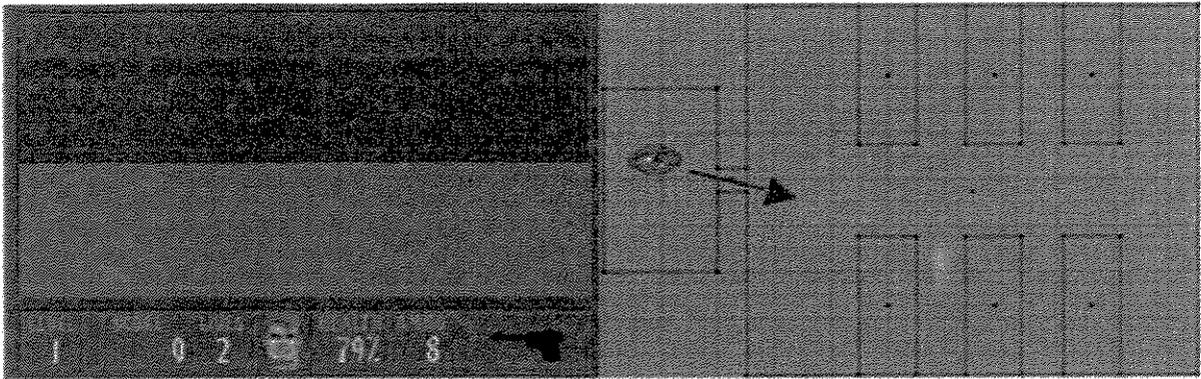
jogador, que podem desencadear outros eventos. O jogador tem liberdade total para exploração do ambiente.

Conforme caminha pela fase, o jogador tem seu campo de visão recalculado e mostrado por um processo de dinâmico que atualiza as informações na tela, várias vezes por segundo, dando a impressão de um movimento contínuo. Este processo chama-se renderização. Renderizar do inglês: to Render: é o processamento final feito após a edição de todos os parâmetros de um objeto de trabalho (modelo 3d, vídeo, áudio multicanal. Etc.), com o objetivo de obter o trabalho em sua forma final. Este processo é usado também quando se cria uma fase no editor de jogos.

O grande avanço de Wolfenstein 3D foi ter conseguido renderizar o ambiente do jogo em tempo real. Este processo consumia muito da capacidade do processador, da memória e da placa de vídeo.

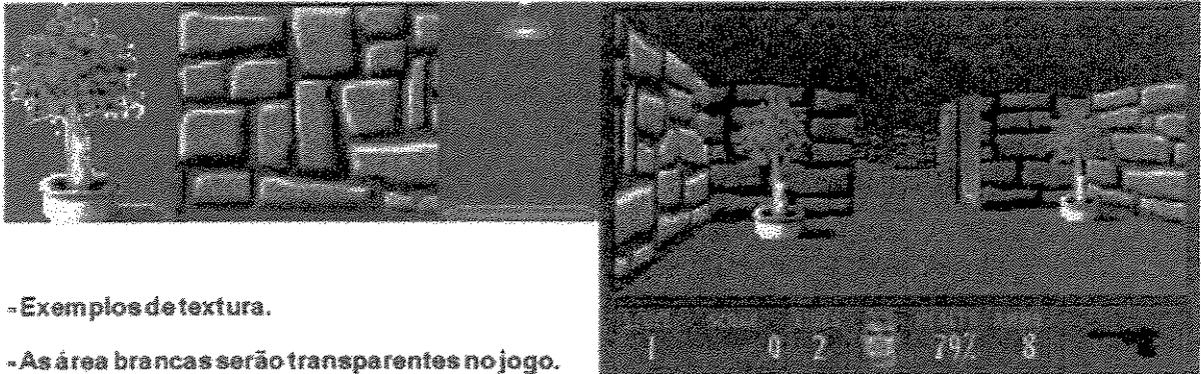
A técnica usado pelo jogo é ilustrada a seguir:

Quadro 1.1 - Os Passos de Renderização em Wolfenstein 3D



- Primeiro o programa "desenha" as partes fixas da interface: o painel de status e o fundo do cenário.
- O chão e o teto da fase são desenhados em cores sólidas, economizando processamento.

- A seguir, ele lê as informações do mapa 2d e calcula o ângulo de visão do jogador.

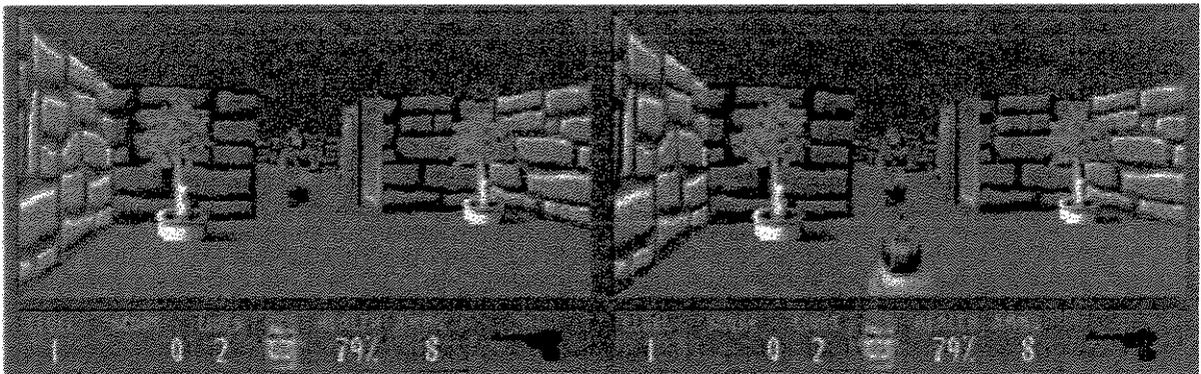


- Exemplos de textura.

- As área brancas serão transparentes no jogo.

- Para cada face do mapa, ele acessa a biblioteca de texturas e "pinta" cada uma com a textura correspondente.

- A seguir, ele lê as informações do mapa 2d e calcula o ângulo de visão do jogador.



- O soldado presente na cena é uma "sprites" dinâmica. Possui versões com pelo menos oito ângulos de visão: lateral direita, lateral esquerda, de costas, etc.

- Por fim, a mão do jogador é desenhada com a arma correspondente.

A indústria de Jogos Eletrônicos Hoje

Para o ano de 2002, a indústria de jogos eletrônicos caminha para um faturamento global de 31 Bilhões de dólares segundo a agência inglesa *Informa Media Group*. Os três maiores fabricantes de consoles: Sony, Nintendo e Microsoft deflagram uma guerra de preços pela Ásia, Europa e Estados Unidos e a demanda é crescente.

As vendas dos três consoles combinados chega a 22 bilhões de dólares num total de 70.1 milhões de unidade instaladas.

Novos mercados estão surgindo com a nova geração de telefones celulares, com a Tv interativa e ainda este ano devem chegar ao montante de \$873 milhões.

A indústria mostra amadurecimento e quer diversificar seu mercado, buscando um público além dos chamados *core gamers* (aficionados) e está investindo em conteúdo e em novos tipos de jogos para conquistar os *casual gamers* (eventuais).

Já a indústria de jogos de computadores deve chegar à soma de 8.5 bilhões de dólares.

O Panorama Brasileiro

O Brasil continua à margem deste processo ainda não tem destes consoles de ponta sendo fabricados em território nacional, mesmo que todas estas empresas estejam estabelecidas no país. Quanto aos desenvolvedores de jogos nacionais, ainda são poucos e com iniciativas isoladas, alguns centros de pesquisas/incubadoras tecnológicas e estúdios estão crescendo e já despertam para este mercado.

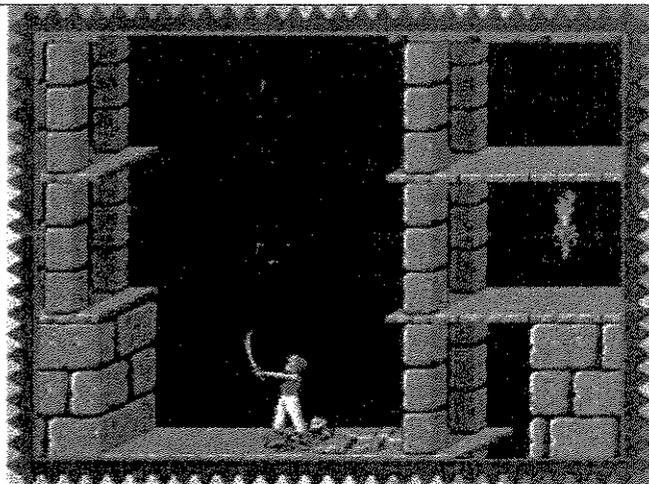
A Universidade Federal de Pernambuco já possui até disciplinas de graduação específicas sobre jogos, recentemente a Continuum Entertainment de Curitiba conseguiu projeção internacional com o seu título : *Outlive*.

Apresentação dos Jogos Analisados

Price Of Persia – Jordan Mechner, 1990.

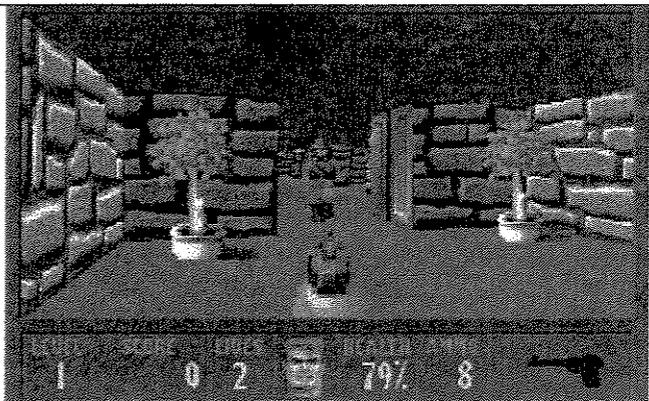
O Príncipe têm uma hora para escapar das masmorras e libertar sua amada das garras do maléfico vizir.

O jogo apresenta uma perspectiva fixa, mas partes do cenário são “desenhadas” em camadas diferentes, proporcionando uma ilusão de um ambiente tridimensional.



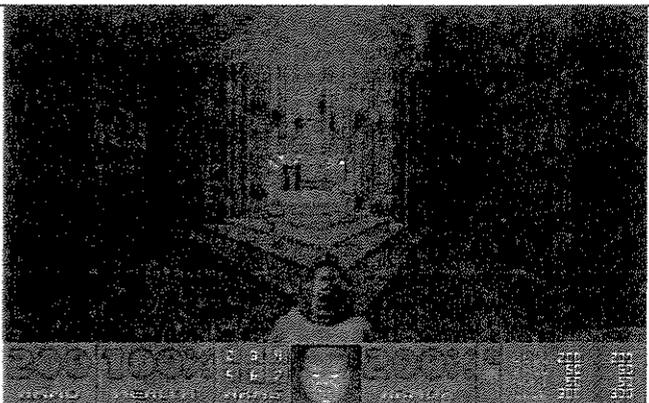
Wolfenstein 3D – Id Software, 1993.

O primeiro jogo de ação em primeira pessoa propriamente dito! Em Wolfenstein, você é um soldado “Aliado” na 2ª Guerra Mundial capturado em um castelo Nazista. Seu objetivo é escapar e durante sua fuga ensangüentada, lutará contra soldados, cães e até o próprio Fúher.



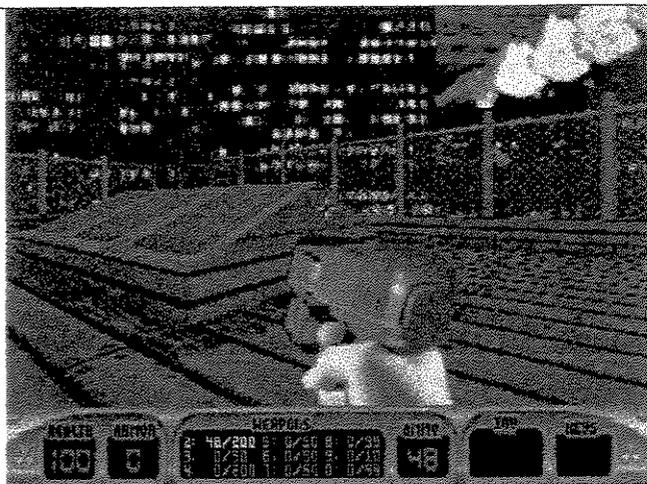
Doom2 – Id Software, 1995.

Continuação do grande sucesso do ano anterior, Doom. Apresenta pela primeira vez cenários que possuem mais de um plano vertical. Marca a confirmação arquétipo do herói no estilo Soldado Futurista (*Space Marine*).



Duke Nuken – 1996.

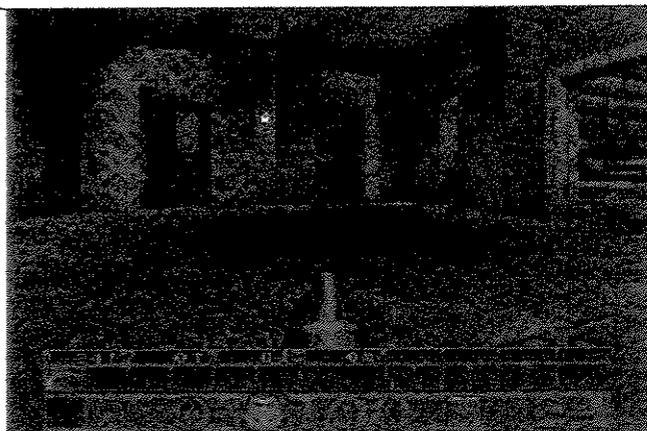
Duke Nuken é o primeiro personagem “real” dos jogos *FPS*, é o perfeito herói hollywoodiano, o boa-pinta, cheio de dinheiro e está pronto para salvar o mundo da invasão de alienígenas.



Quake – Id Software, 1996.

Novamente o Space Marine deve salvar a terras de criaturas bestiais oriundas dos mais profundos infernos.

É o primeiro jogo que permite ao jogador olhar/atirar com liberdade nos 3 eixos (x,y,z).



Quake 2 - Id Software, 1997.

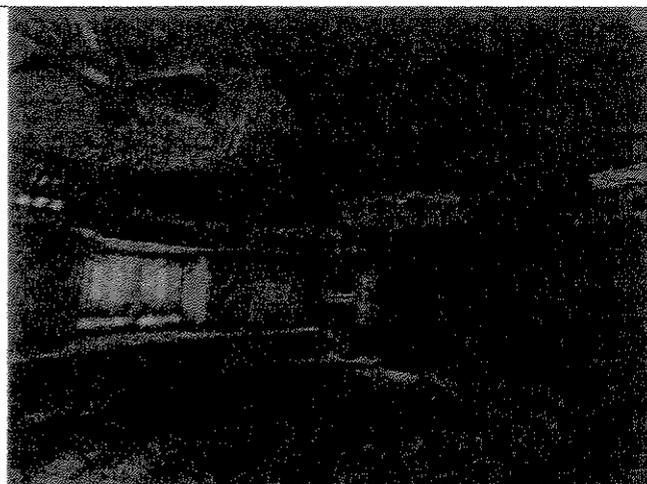
Quake 2 foi um jogo tão ruim que só está nesta lista devido à flexibilidade de seu *engine* que permitiu que fossem feitos outros jogos de peso como: Half-Life, Counter-Strike e Vampire – The Masquerade - Redemption



Unreal – 1997

O jogador é um prisioneiro que está sendo removido para um planeta prisão nos confins do universo, quando a nave sofre um acidente e ele se vê em uma fuga desesperada em um planeta hostil.

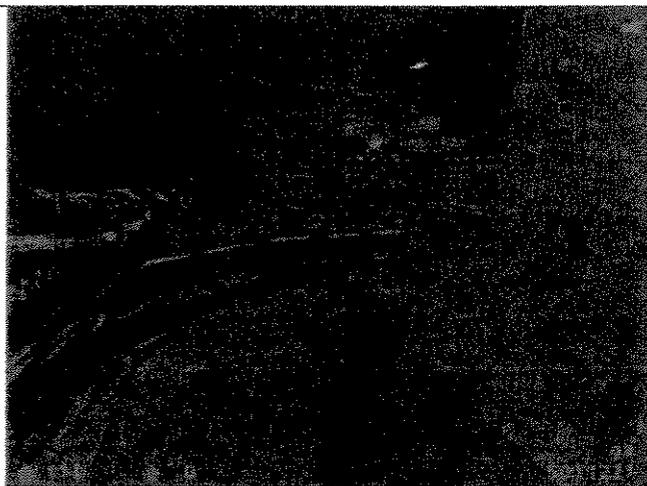
O visual do jogo marca época do padrão 3Dfx



Half-Life – 1997

O físico Gordon Freeman está em um experimento em uma base de pesquisa secreta do governo quando algo dá errado.

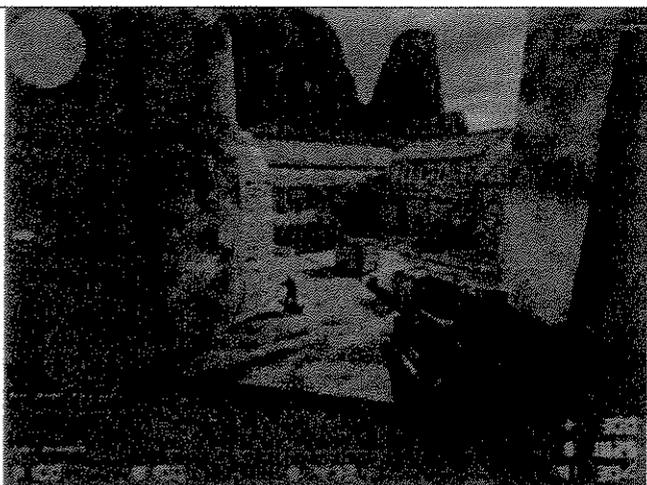
Com um enredo elaborado e com um design de ambientes primoroso, é um dos títulos obrigatórios para quem se interessa por jogos de computadores.



Counter-Strike – 1998

É uma modificação feita sobre a plataforma de Half-Life em que dois times: terroristas e forças anti-terroristas se enfrentam em jogo de estratégia e habilidade.

É o jogo multijogador mais jogado atualmente no mundo.



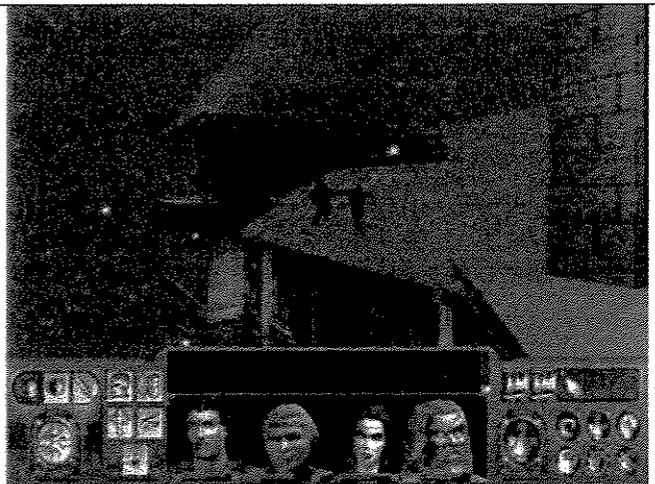
Quake 3 Arena - Id Software, 1999.

A arena virtual de combate onde o único enredo é atirar para matar.



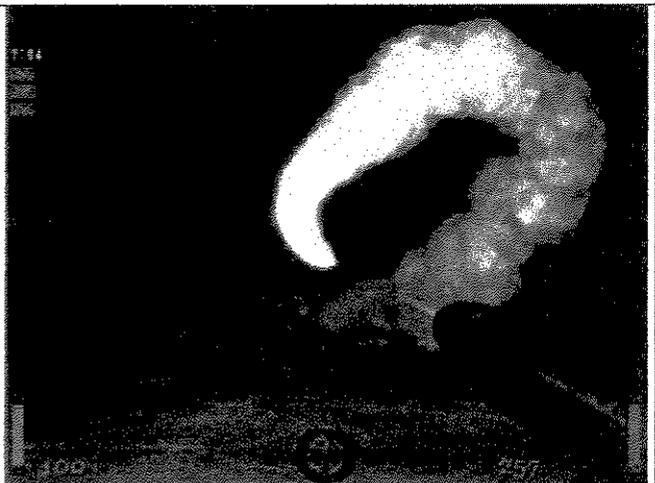
Vampire - The Masquerade: Redemption, 2000.

Adaptação do RPG homônimo, onde o jogador encarna um vampiro e sua busca pela salvação da alma de sua amada. O enredo transcorre durante oitocentos anos e dois continentes.



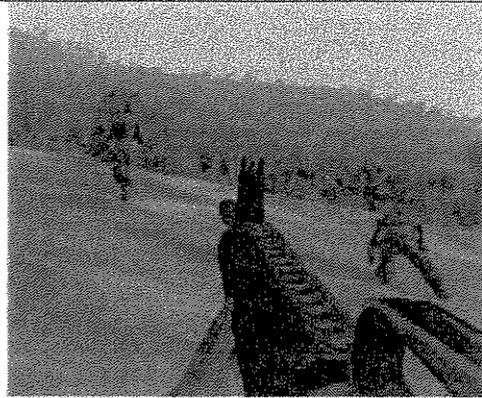
Return to Castle Wolfenstein - Id Software, 2001.

Mais uma vez os Aliados têm de enfrentar as terríveis armas de guerra nazistas.



Operation Flashpoint – 2001.

Um simulador de soldados que se passa durante e a guerra fria e foi destaque no festival dos jogos independentes de 2002.



Capítulo 2 – O Jogo Black Death

O Início do Projeto

Um dos grandes problemas na criação de um jogo é o seu ponto de partida, pois assim como um filme, ou um espetáculo teatral, o jogo possui uma estrutura que se assemelha a uma dramaturgia, ou a uma narrativa, ainda que não seja propriamente nem uma nem outra. Por trás do jogo há uma história da qual o jogador participa. Encontrar uma boa história é a primeira etapa para a criação de um jogo.

A Idéia Inicial

Foi numa conversa informal que um amigo falou-me sobre o livro *The second Messiah*⁹; o romance narra o poder e o conhecimento que os cavaleiros Templários detinham na época das Cruzadas.

O autor levanta a hipótese que os templários depois de tomarem posse do Templo de Salomão em Jerusalém, teriam recolhido e guardado em segredo os pergaminhos dos Essênios, tidos como os “professores” de Jesus Cristo. Todos estes documentos teriam sido salvos do grande massacre de templários promovido por Felipe II, o Belo (França) e teriam sido levados pelos cavaleiros sobreviventes até Portugal e depois para a capela de Rosslyn na Escócia. Sob a proteção do Rei Robert the Bruce, a Escócia foi o único país que não executou as ordens do Papa que extinguiu a ordem dos Templários. Tornando o país uma espécie de santuário dos Templários remanescentes.

Romances e lendas como essas são muito úteis para a criação de um jogo. Os fatos históricos servem somente como pano de fundo para o desenrolar do jogo, o referencial para o “argumento” do jogo é de caráter romanesco e não necessariamente precisa ter compromisso com o fato histórico. Daí o caráter de arte popular do jogo eletrônico, como exalta o pesquisador do MIT, Henry Jenkins¹⁰.

O principal objetivo de um jogo é o entretenimento, qualquer informação adicional que o jogador receba durante o jogo são muito bem-vindas, mas este não é o seu intuito.

⁹ *The second Messiah; Templars, the Turin Shroud and the Great Secret of Freemasonry*. Christopher Knight e Robert Lomas. Harper Collins, Reino Unido, reedição, 2000.

¹⁰ Em Jenkins, Henry. *Art Form for the Digital Age*, Technology Review, Setembro/Outubro de 2000. MIT Press.

Templários e Peste Negra

Usar um Cavaleiro Templário como o personagem principal parecia bastante interessante, mas um grande herói precisa de um grande desafio! Foi aí que surgiu a idéia de localizar a história durante a “Peste Negra”. Todo o percurso da Peste através do oriente até a Europa medieval, os fatos e personagens históricos, proporcionavam uma história com vários caminhos e conseqüências. Transformar o enredo em jogo era um desafio bastante atraente, cada trecho da viagem do bacilo (responsável pela Peste) até a Europa podia gerar uma fase. Por exemplo: a Batalha em Caffa, ou a viagem dos navios genoveses contaminados até Gênova, ou

A embrião de um roteiro se formara.

Argumento

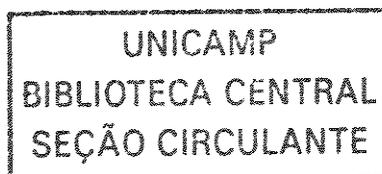
Ásia Menor do Séc. XIV, ano de 1348. Antes da Peste Negra.

O personagem principal (o Jogador) é um “ex-cavaleiro” Templário que teve contato com os mongóis e presencia a Batalha de Caffa, na Criméia. Essa cidade era um entreposto comercial genovês na região. Foi durante essa batalha que os mongóis, pensando que os genoveses eram os culpados pelo flagelo que os assolava, decidiram catapultar seus soldados mortos pela peste bubônica para dentro da cidade.

Os genoveses, ao perceberem as dimensões daquela catástrofe, abandonaram a cidade e retornaram para a Europa. Aos poucos, os passageiros dos barcos começaram a cair enfermos, logo todos perceberam que aquela viagem não chegaria a seu fim. Por todos os portos por onde passaram, estes barcos não tiveram permissão de atracar. Porém os ratos que infestavam os porões chegaram à terra com sua “carga mortal”.

O nosso Herói, conhecendo a terrível peste que aqueles navios portavam, parte para a Europa, tentando impedir que aquelas embarcações cheguem a terra firme. Vai iniciar uma empreitada contra a peste, lutando contra o inevitável.

Em três meses, todos principais portos da Europa estariam infectados. Até 1351, morreram vitimados pela Peste Negra aproximadamente 20 milhões de pessoas no continente Europeu.



As Fases do Jogo – ou o Desenrolar do Enredo

Num esboço inicial, para contar a história do templário contra a Peste Negra imaginei que seriam necessários pelo menos 5 cenários e dois vídeos:

- Prólogo: Vídeo da Batalha de Caffa

O Templário Hugh segue para Caffa ao saber que a terrível doença chegara a cidade. Lá encontra a cidade quase em ruínas, casas em chamas e cadáveres de guerreiros com manchas negras e inchaços espalhados por toda a extensão de seus corpos. A batalha entre os mongóis e os genoveses ainda continua e os soldados remanescentes, com suas catapultas, arremessam os corpos de seus companheiros para dentro do último reduto genovês fortificado. Ao contornar o forte, percebe que várias embarcações genovesas acabavam de partir rumo à sua terra de origem.

Imediatamente segue para o porto. Sua jornada continuaria pelo mar.

- A Chegada a Gênova

Viajando sozinho e sem descanso ele consegue chegar à praia de Gênova. Ele precisa encontrar seu mestre e antigo companheiro de batalha, o monge Foucault, para que juntos possam impedir que os barcos vindos de Caffa aportem. O monge se esconde na Catedral e para chegar até lá Hugh usará o atalho pelas galerias de águas fluviais. Construídas em várias épocas diferentes, as galerias são um grande labirinto para aqueles que não conhecem os símbolos templários.

- A Catedral

Hugh chega aos subterrâneos da Catedral, ela é formada por vários níveis de túneis e corredores e esconde várias construções que serviram às atividades templárias. Mais que uma prova de habilidade, a Catedral é uma prova de fé para o templário.

- Lanterna de Gênova

É noite, para chegar a Lanterna, o cavaleiro terá de passar soturnamente pela cidade. Seu objetivo é apagar a pira que arde no topo da torre e impedir a chegada dos barcos

vindos de Cafa. Terá de enfrentar os guardas da Torre. As embarcações genovesas se aproximam.

- Fortaleza de Tomar em Portugal

A fortaleza de Tomar é um dos últimos redutos dos templários, lá eles deverão reunir os últimos templários e traçar um plano de ataque contra a Peste. Em meio à reunião os templários são surpreendidos pelo ataque dos cavaleiros teutônicos. Hugh e Foucault são capturados e levados até Avignon.

- Avignon

Lá o rei da França confessa ter conspirado com teutônicos e subornado os mongóis para que infectassem Cafa pois sabia que os templários restantes apareceriam. Assim ele poderia caça-los e conquistar os tesouros guardados pelos templários que pertenciam ao templo de Salomão em Jerusalém, entre eles estariam a Arca da Aliança e talvez o próprio Graal!

Epílogo: Vídeo Final ...

Hugh e Foucault estão amarrados e prontos para serem queimados em uma grande fogueira no pátio do palácio, quando os cavaleiros teutônicos chegam aos gritos, dizendo que a Peste alcançara a cidade. Em meio ao tumulto da multidão correndo desesperada, os templários fogem...

Como se Faz um Jogo Eletrônico

Não existe jogo sem uma mídia e suas respectivas ferramentas. Sem pedras, cartas, computadores ou até mesmo a linguagem, não se pode criar ou jogar nenhum jogo. Genericamente, pode-se dizer que o jogo é um conjunto de regras que definem objetivos, as formas de alcançá-los, determinando a derrota ou a vitória dos jogadores.

Tratando-se de um jogo eletrônico, deve-se escolher as ferramentas corretas, pois cada tipo de programa serve a propósitos diferentes e conseqüentemente leva a resultados diferentes. Este tipo de escolha é crucial para o resultado, a equipe de desenvolvimento (designers, programadores, artistas, produtores, etc.) tem de avaliar qual tipo de tecnologia lhe dará mais flexibilidade, longevidade e qualidade no produto final.

Na primeira versão deste projeto em 1999, estudou-se a possibilidade de uso de softwares de orquestração¹¹ em multimídia. A escolha seria por uma plataforma comum no mercado brasileiro como Authorware ou Director, ambos da Macromedia. Esses programas permitem que se combinem imagens *bitmap*¹² como: fotos, mapas, animações, sons e textos em um multimídia interativo.

Com noções básicas de programação, um usuário de computador pode fazer desde uma apresentação multimídia, como integrar seu programa com um banco de dados, ou executar outros aplicativos durante sua apresentação, criar programas de treinamento e simuladores.

Esse tipo de ferramenta é fácil de ser usada. O programador multimídia não precisa digitar linhas de código ou criar scripts. Sua principal tarefa neste tipo de ambiente é cuidar da arquitetura da informação e do design da interface. A maior parte da programação no Authorware é feita em uma interface gráfica, o usuário combina os ícones referentes ao tipo de conteúdo (áudio, vídeo, texto, etc.) que deseja utilizar e os coloca no local e ordem que este objeto deve aparecer ou ser acessado.

¹¹ Orquestração é um termo cunhado pelos produtores de mídias interativas, consiste no processo de organizar os vários eventos e elementos da apresentação multimídia. É dizer onde e como serão mostrados os gráficos, textos, sons, etc. Apesar de Orquestração ser um termo técnico da Música, seu sentido na Multimídia não corresponde ao conceito musical.

¹² As imagens *bitmap* também chamadas de imagens rástreas são matrizes de pontos, onde cada ponto contém informações de posição e cor. Os formatos mais comuns são: .jpg, .gif, .tiff, .bmp.

O usuário tem como resultado de sua programação um fluxograma com toda a navegação que acabou de criar. Assim, ele vê como toda sua navegação está organizada, reconhece quais os tipos de objetos estão acessados ou mostrados a cada momento de seu multimídia.

Authorware foi a primeira escolha, apesar de ser uma ferramenta poderosíssima, não é um programa específico para a criação jogos de ação em primeira pessoa. Ele só gerencia imagens *bitmap* e não *renderiza* nenhum conteúdo gráfico. Isto equivale dizer que qualquer ilusão de se estar em um ambiente 3D é dada pelo uso de uma foto em perspectiva já pronta, como nos jogos de interface *point-and-click* (ver cap. I).

A Figura 2.1 traz as visões da interface de trabalho do Authorware e a interface do multimídia pronto. Em (a), vemos a interface de edição de conteúdo do programa. Os ícones organizados em forma de "árvore" representam a navegação da apresentação multimídia. Em (b), temos o ícone "fundo 1" que contém a foto de fundo do multimídia. Em (c), vemos os ícones que representam os objetos que funcionarão como botões. Em (d), temos a interface pronta do multimídia. Em (e), temos o resultado da seção em sua forma final.

Authorware - exemplo de orquestração e produto final

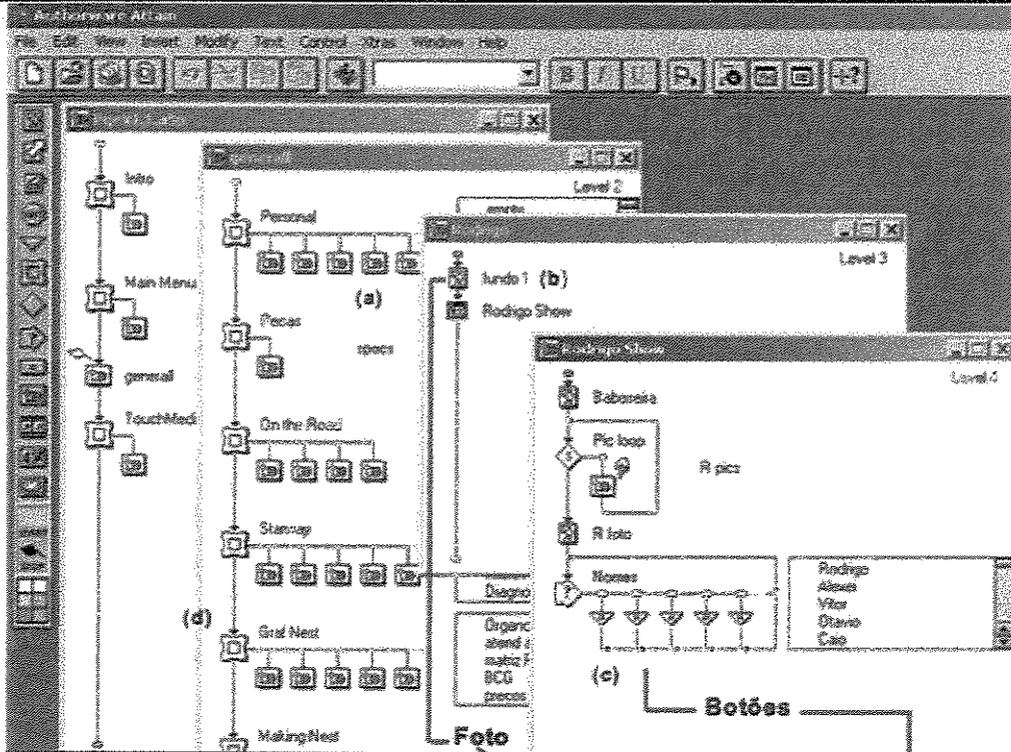


Figura 2.1 - Authorware - exemplo de orquestração e produto final

Uma Nova Estratégia

A partir do final da década de 90, alguns dos fabricantes de jogos para computadores adotam uma nova estratégia. Visando conquistar e cativar um público mais amplo, decidiram distribuir juntos com seus títulos de peso, seus respectivos programas de edição. Estes programas recebem o nome genérico de “Editor” e colocaram ao alcance do público as mesmas ferramentas usadas pelos designers de jogos na produção desses títulos.

Logo, surgiram na Internet as comunidades de fãs que contribuíam com novas fases, novos *skins* (texturas para personagens) e até com modificações do jogo original, os *Mod`s* (*Modification*). O ciclo de vida do jogo contava agora com as contribuições ativas dos jogadores de todo o mundo.

O caso mais famoso de Mod é Counter-Strike (Minh Le, 1998) onde dois times: Terroristas e Forças Anti-terroristas disputam com objetivos e armas distintas. Hoje é o jogo mais jogado em rede em todo o mundo e foi responsável pela multiplicação de um novo ramo de comércio, as LAN Houses (casas para jogos em rede).

As “Engrenagens” de um Jogo de Computador

A estrutura dos jogos atuais podem ser divididas em duas esferas distintas: a esfera **Tecnológica** e a esfera **Estética**.

A **Esfera Tecnológica** reúne todos os elementos que controlam a execução do jogo. O principal deles é o **Engine** (motor), o programa que executa o jogo propriamente dito. Dependendo da arquitetura do *Engine* ele pode vir acompanhado de vários outros arquivos como DLL (Dynamic Library Link – no caso de sistema operacional Microsoft Windows), Scripts, etc.

O *Engine* tem as seguintes funções:

- definir o ângulo e a perspectiva de visão do jogador (primeira pessoa, terceira pessoa).
- “conversar” com o processador e com todos os dispositivos de entrada e saída: placa de vídeo, placa de som, Joystick, mouse, etc.

- controlar os personagens animados ou atores (não controlados pelo jogador)
- Comandar elementos de cenários, efeitos de iluminação
- Capturar, interpretar as ações do jogador através de sensores¹³
- Responder a estas ações de acordo com sua programação.
- Comandar a renderização de áudio e de vídeo em tempo real.

Enfim, O *Engine* controla todos os eventos que compõem a ação do jogar.

O *Engine* é um programa definido, vendido como um produto acabado. Quando uma empresa decide criar um jogo a partir de uma plataforma existente, é pelo *Engine* que se paga a licença do software. Em termos de programação, ele é feito geralmente em uma linguagem orientada a objeto, a mais usual é a C++. Este centro do programa é por sua vez, é controlado por objetos programados em níveis mais altos¹⁴ como **classes**¹⁵ e **scripts**.¹⁶

Devido a complexidade de suas tarefas, ele é o programa mais caro no desenvolvimento do projeto de um jogo de computador. A escolha de se criar ou comprar um *engine* pronto é um passo crucial, pois implicará nas limitações de uso, nos custos de desenvolvimento, incluindo não só a equipe de programadores, como também a equipe de designers, artistas e programadores.

Já a **Esfera Estética**, é aquela regida pelo conteúdo: Roteiro, direção de Arte, Design Visual e Sonoro. Reflete todas as opções estéticas da equipe. Sempre é bom salientar que estas duas esferas devem sempre se entrecruzar, pois uma condiciona a outra. É nesta esfera que os grandes jogos se diferenciam. Na forma de mostrar o roteiro

¹³ Sensores – Dentro do jogo existem vários elementos que capturam e monitoram toda a ação do jogador e que por meio de programação específica direcionam as respostas do *Engine*. Este tópico é visto com maior profundidade na seção **Interação** deste mesmo capítulo.

¹⁴ Em computação os programas passam por vários níveis de interação desde de o processador até o usuário. O nível “mais baixo” de interação são as instruções enviadas diretamente ao processador enquanto as interfaces gráficas, por exemplo, são consideradas de “alto nível” pois estão mais próximas da interação do usuário.

¹⁵ Classe: grupo de objetos com as mesma qualidades.

¹⁶ Scripts são parâmetros escritos em forma de texto que comandam determinados objetos.

(no desenrolar da trama), no mecanismo de fantasia, ou melhor, em qual fantasia ele imerge o jogador e como o faz alcançá-la.

Não basta fornecer as armas mais barulhentas e cheias de efeito e colocá-lo em uma arena de tiro. Um grande jogo consegue a imersão e a participação do jogador, quando os elementos usados são convincentes e coerentes. O jogador transforma-se em um cúmplice/ator do mundo criado pelos designers. No ato de jogar deve haver sempre o prazer da descoberta, do desafio a ser vencido, na vitória ou na derrota.

O Editor de Jogos

O editor de jogos é o programa que a partir de arquivos padrões de imagens, áudio, modelos tridimensionais e vídeos, gera arquivos compatíveis o *engine* ou motor.

Ele organiza todos conteúdos utilizados pelo *engine* em bibliotecas específicas: ex. bibliotecas de texturas (imagens), de efeitos sonoros, de músicas, de animações, objetos, classes, etc.

Geralmente, este programa é feito pela empresa que cria o conteúdo do título (novo jogo). Um mesmo *engine* pode ter vários editores, criados por empresas diferentes. Por exemplo, a Id Software, criadora do *engine* de Quake e Quake2 tem seu editor chamado QRadiant, já a Valve Software criadora de Half-Life, desenvolveu um editor próprio Hammer. Este fato se deve às especificidades de cada projeto. Para exemplificar, citamos dois jogos que usam o *engine* de Quake2, mas criados em seus editores próprios:

- **Half-life** (Valve Software, 1997) é um jogo com várias seqüências “cinematográficas” onde o jogador assiste a uma conversa entre dois personagens, ou é alertado sobre um perigo adiante. Estas seqüências não impedem que o jogador se movimente livremente, não interrompem a ação de jogar. O jogador é convidado a participar daquela cena e sabe que ela lhe dará mais informações para seguir em sua empreitada. Conseqüentemente, o editor desenvolvido pela Valve software tem funções específicas para a criação destas seqüências através de *scripts*.
- **Vampire: The Masquerade – Redemption** (Nihilistic Software, 2000) é a adaptação para pc do jogo de RGP homônimo. Em seu modo *single player*, também usa várias seqüências “cinematográficas”, mas o jogador só assiste aos outros personagens. Em compensação em seu modo multijogador, possui uma ferramenta muito eficiente para a criação de novos enredos. O modo *Storyteller* (narrador) permite que o jogador comande toda a trama de jogo em tempo real. O grande mérito da Nihilistic, foi criar uma ferramenta que permite que várias outros enredos possam ser jogados em seus cenários.

Na Figura 2.2 temos dois exemplos de editores de jogos comerciais: Worldcraft (Valve Software) e Q3Radiant (id Software). Apesar de pertencerem a fabricantes diferentes os programas apresentam pontos comuns na interface. Em (a) e (a'), estão as janelas para a visualização do ambiente tridimensional. Em (b) e (b'), temos a seção para a edição das geometrias do ambiente. Em (c) e (c'), vemos as janelas para visualização das bibliotecas de texturas.

Na Figura 2.3 apresenta duas cenas tomadas dos jogos Half-life e Vampire: The Masquerade – Redemption (VTMR). Em (a), vemos uma seqüência em script de Half-Life onde os personagens controlados pelo computador saúdam a chegada do jogador a ao Centro de Pesquisa "Black Mesa". O guarda e o cientista informam que houve uma pane nos computadores e por isso estão com dificuldades de comunicação com o mundo exterior. Em (b), temos uma foto de uma seção mutijogador de VTMR onde a *grande cabeça* a frente do grupo de jogadores representa o narrador (*storyteller*). O narrador comanda inserção de objetos, inimigos, atribui poderes e prêmios aos jogadores. O ícone do Storyteller é visto somente na máquina que hospeda o jogo.

Exemplo de Editores: Hammer e Q3Radiant

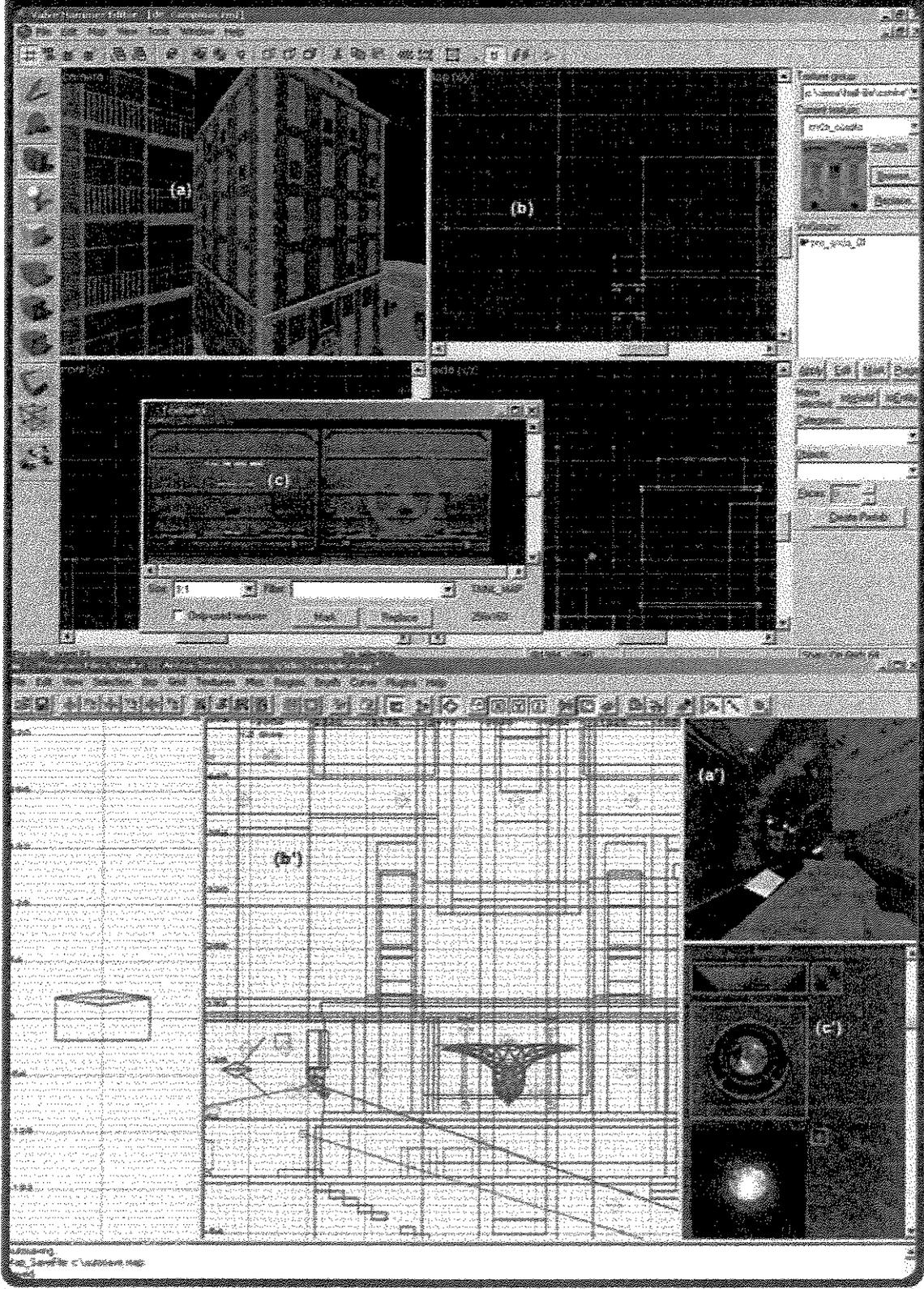


Figura 2.2 - Exemplo de Editores: Hammer e Q3Radiant

Dois jogos, um Engine: Half-Life e Vampire - The Masquerade: Redemption



(a)



(b)

Figura 2.3 - Dois jogos, um Engine: Half-Life e Vampire - The Masquerade: Redemption

UnrealED – Apresentação e Características

O UnrealEd é o editor do jogo Unreal da Digital Extremes/ Epic Games. Ele é distribuído gratuitamente junto com o jogo e conta com várias “comunidades” de desenvolvedores, artistas e aficionados na Internet. Esta estratégia reflete o interesse das indústria de jogos em formar profissionais e atrair público para seus produtos. Sem dúvida, a Internet é quem melhor atende às necessidades de trocas de informações e sobre esta tecnologia em constante desenvolvimento.

O programa é bastante flexível e robusto e sua arquitetura é composta por um núcleo feito em C++, várias *classes* feitas em Java que são controladas por *scripts* escrito em uma linguagem própria chamada UnrealScript. Conhecimentos de Linguagens Orientadas a Objeto, principalmente Java e JavaScript são aconselhados pelos fabricantes para aqueles que desejam desenvolver em seus projetos usando este editor.

O UnrealEd não é editor para modelagem detalhada de objetos e personagens como o 3D Max da Autodesk ou MAYA da Alias Wavefront, ele só possui quatro primitivas¹⁷ (cubo, esfera, cilindro e cone) e só realiza operações de soma e subtração de volumes.

O programa tem uma característica peculiar em seu projeto: ele trata o ambiente 3D inicial (*world*) como um grande bloco maciço. O processo de criação de fases no UnrealEd pode ser comparado a uma escultura às avessas, como se escultor inicia-se sua obra de dentro para fora, retirando os blocos da pedra.

Outro fator importante do programa é a forma como ele trata as texturas. As texturas são imagens *bitmap* que revestem os objetos tridimensionais. O UnrealEd só trabalha com textura com dimensões em pixels¹⁸ em potências de base 2 (ex: 16x32; 64x64, 128x128, etc).

¹⁷ Os Modeladores profissionais trabalham com formas complexas, com dezenas de milhares de polígonos. Servem para animações complexas de personagens e objetos como no filme “Toy Story” da Pixar Production. Os editores de jogos servem para integrar os polígonos que formam o ambiente do jogo e o ambiente de programação.

¹⁸ Pixel - Abreviatura de “picture element” (elemento de imagem). Os pixels são pontos em uma tela de computador ou televisão que são combinados para formar uma imagem. As imagens de computador são criadas como uma matriz de pixels, cada uma tendo uma cor específica.

Na Figura 2.4 temos um exemplo de como são organizadas as classes no UnrealEd e como funciona o UnrealScript de um *trigger*.

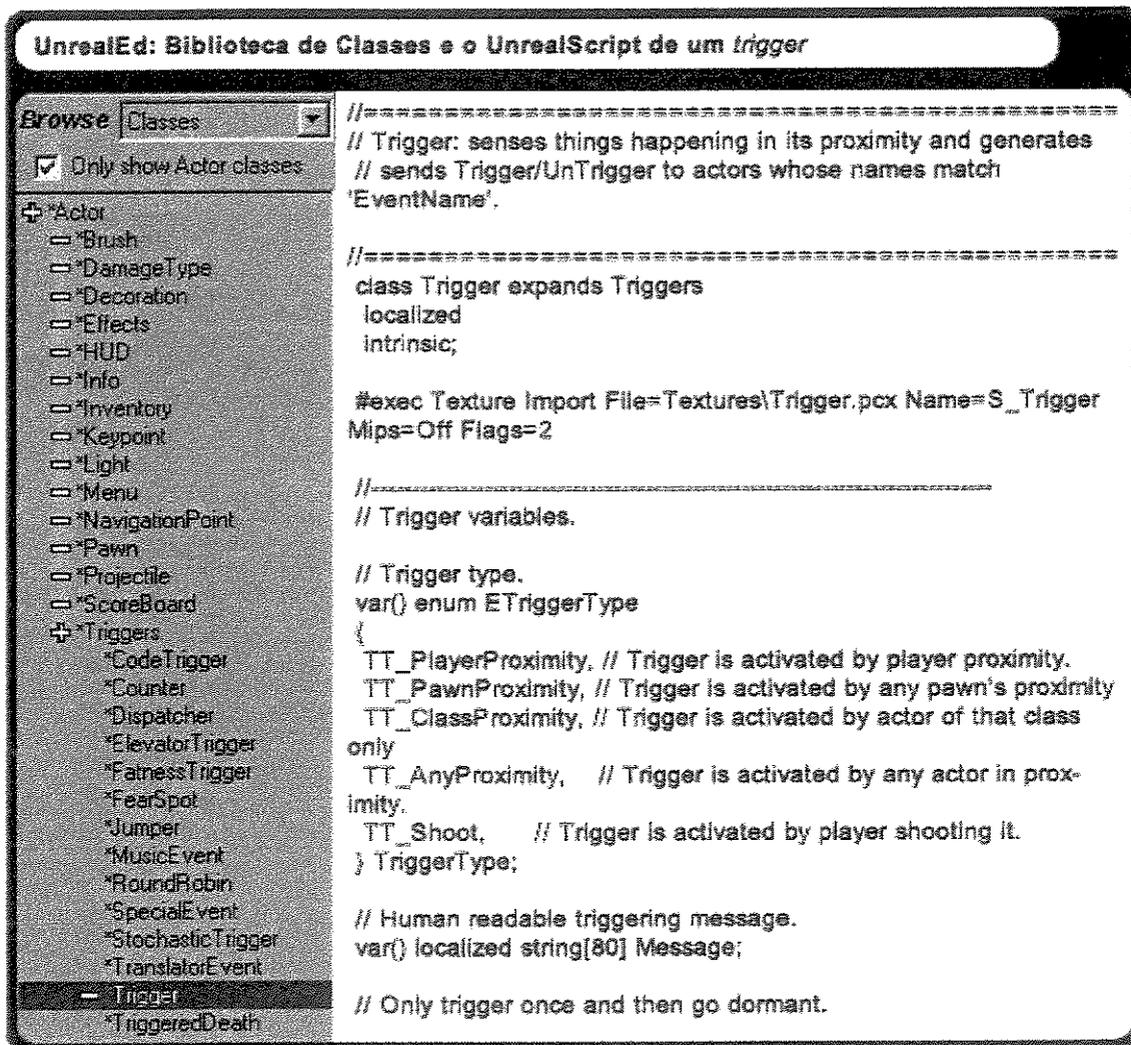


Figura 2.4 - UnrealEd: Biblioteca de Classes e o UnrealScript de um trigger

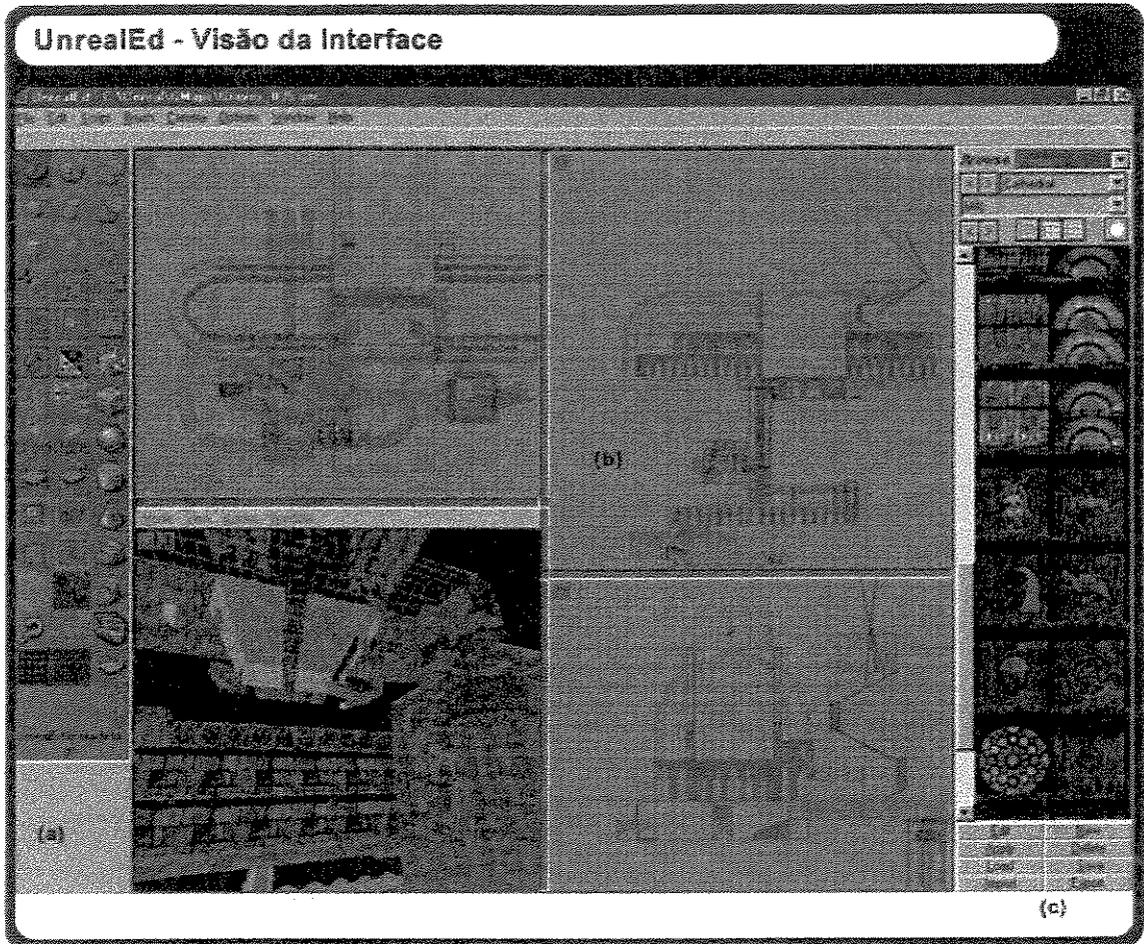


Figura 2.5 - UnrealEd - Visão da Interface
(a) Botões para edição de volumes e formas.
(b) Câmeras para visualização
(c) Área para navegação de bibliotecas: Texturas, Classes, Sons e Música.

Black Death

O jogo **Black Death**, objeto deste trabalho, é o que se chama usualmente de um **Mod**, do inglês *modification* (modificação). Este termo indica que Black Death não é um jogo novo propriamente dito e sim um produto modificado, sem fins comerciais, que usa produtos e tecnologias já prontas e devidamente registradas, no caso: o jogo Unreal e editor UnrealEd.

Tendo em vista a dificuldade de design e programação do jogo completo, optou-se por realizar apenas a fase da Catedral durante o período do mestrado. Julgamos que ela seria suficiente para alcançar todos os objetivos deste trabalho.

Para criação da fase da Catedral foi criado um “roteiro” para ser usado como base de orientação para design e arte. Como estava trabalhando sozinho não precisei criar um documento muito detalhado¹⁹, descrevendo a fase somente em linhas gerais. A seguir a “ficha de design” da Catedral.

Fase: Catedral

Objetivo: Encontrar o Padre “Foucault” para juntos iniciarem a empreitada.

Considerações para o Design: O jogador está dentro de um tumba falsa e enxerga através das fendas na parte posterior da câmara mortuária. Ao tentar “empurrá-la”, esta quebra-se e ele se vê dentro de um salão imenso com estatuas piras “mágicas”. Ao norte pode-se ver uma porta com o símbolo dos templários. Para sair da sala da tumba ele terá de seguir os passo de um templário e prestar reverência ao mestre.

O jogador tem que perceber que os templários construíram um templo de encontro debaixo da catedral, ela na verdade é apenas uma fachada. As catacumbas mostram uma atividade anterior naquele local, talvez um lugar de grande importância na rede de controle templário. Os subterrâneos são intermináveis e sombrios abandonados. A

¹⁹ Os produtores e designers ainda não tem uma metodologia específica para a criação de um jogo eletrônico. O desenvolvedores têm usado vários métodos emprestados da Engenharia de Software outras áreas e ciências. Algumas proposições podem encontrados em Kreimeier, Bernd. The Case For Game Design Patterns. Gamasutra. 13 de março, 2002.

Catedral ainda está em construção, vários corredores estão impedidos. Será necessário andar pelas “paredes”, pelo “teto”

Descrição

O Estudo de arquitetura e história mostra que a base de toda catedral deste período é feita sobre arcos. Histórica e tecnologicamente os arcos românicos precedem os arcos góticos. Portanto, o uso de arcos românicos para as catacumbas e subterrâneos e arcos góticos (em forma ogival) para a catedral era coerente.

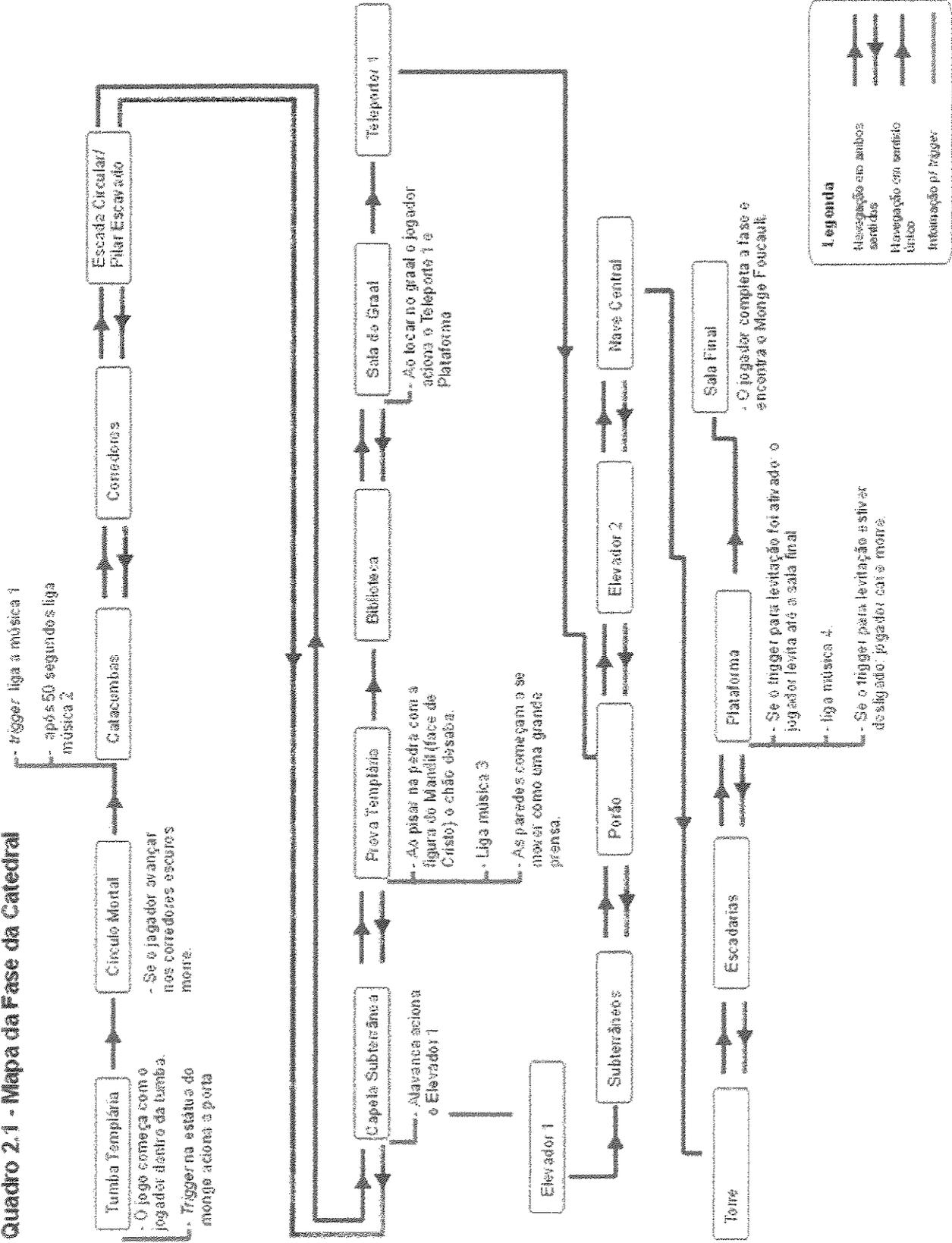
A Catedral foi dividida em 7 grandes áreas

- Cripta Templária
- Catacumbas
- Corredores
- Capela Subterrânea
 - o Prova templária
 - o Biblioteca
- Subterrâneos
- Nave Central
- Torre

Cada área teria arquitetura, sonorização, iluminações característica, com a música permeando certos ambientes. A idéia de seqüência/conseqüência deveria ser passada ao enquanto o jogador percorresse cada área. As “pistas” de onde ir e como funcionam as “coisas” são fornecidas através da luz, da música e das formas.

O Quadro1 apresenta um mapa da fase Catedral. Ele contém as regiões que constituem a “espinha dorsal” da fase, seu fluxo de navegação e algumas informações para os *triggers*.

Quadro 2.1 - Mapa da Fase da Catedral



Criação de Ambientes Virtuais

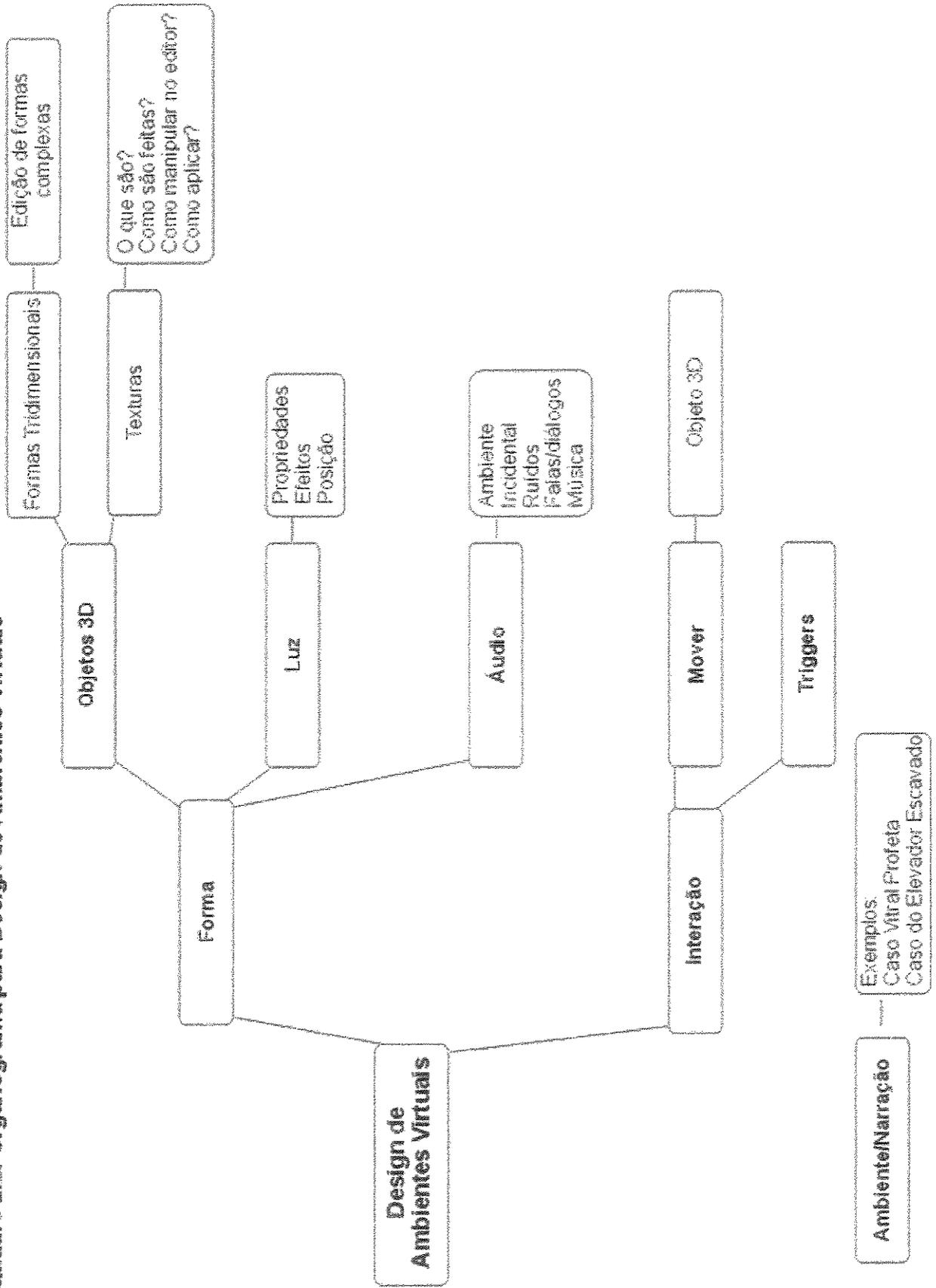
Um ambiente virtual é a morada do jogador por algumas horas. Deve envolvê-lo com sua forma, sua luz, seus sons, seu cheiro... Transformar a idéia do designer em algo que possa ser jogável envolve o trabalho de várias frentes. Gostaria de dividi-las em duas grandes áreas: a da **forma** e a da **interação**.

A **forma** corresponde à infra estrutura da fase, são os elementos de cenário: objetos 3D, luzes e sons.

A **interação** são os objetos que respondem à ação do jogador e sempre envolvem uma instância de programação.

O esquema é apresentado no Quadro 2.2.

Quadro 2.2: Organograma para Design de Ambientes Virtuais



Objetos 3D

O objeto 3D tem como esqueleto um conjunto de vértices com suas posições espaciais e suas propriedades. O *engine* interpreta estes pontos e os transforma em polígonos. O polígono é a base de cálculo de todo o processo daqui em diante, do *engine* até a visualização do jogo pronto na tela do jogador. Um conjunto de polígonos ordenados dão origem a uma forma 3D e é sobre esta forma que geralmente as interfaces dos editores trabalham.

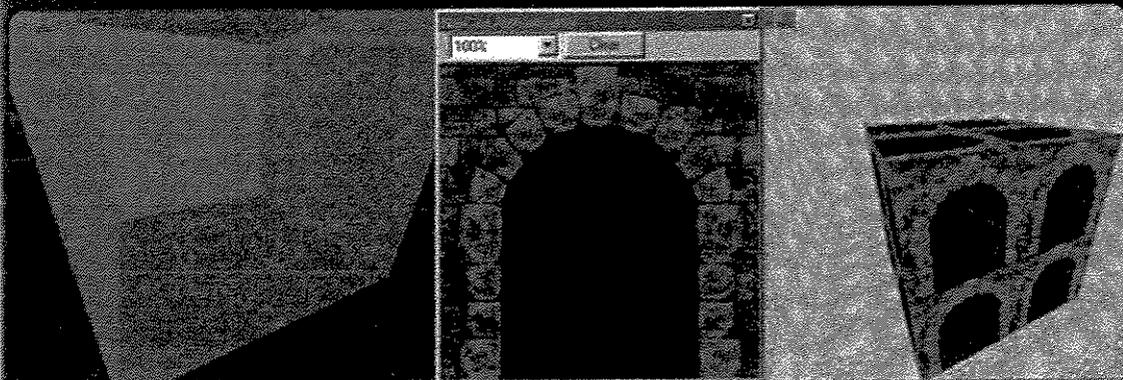
Sobre Formas 3D “Complexas”

A primeira dificuldade de quem inicia-se uso de programas de edição 3D é imaginar como são feitas as formas mais complexas como arcos, cavernas e personagens. O UnrealEd só trabalha com soma e subtração de formas primitivas. É uma ferramenta indicada para edição de objetos simples. Mas, formas mais complexas podem ser feitas a partir de uma seqüência de soma e subtração de volumes.

Dentro deste princípio de soma e subtração está implícito outro fator importante do programa: a qualquer objeto é associado um número para a ordem de renderização.

Os passos para a criação de um arco românico no UnrealEd são visto na Figura 2.6

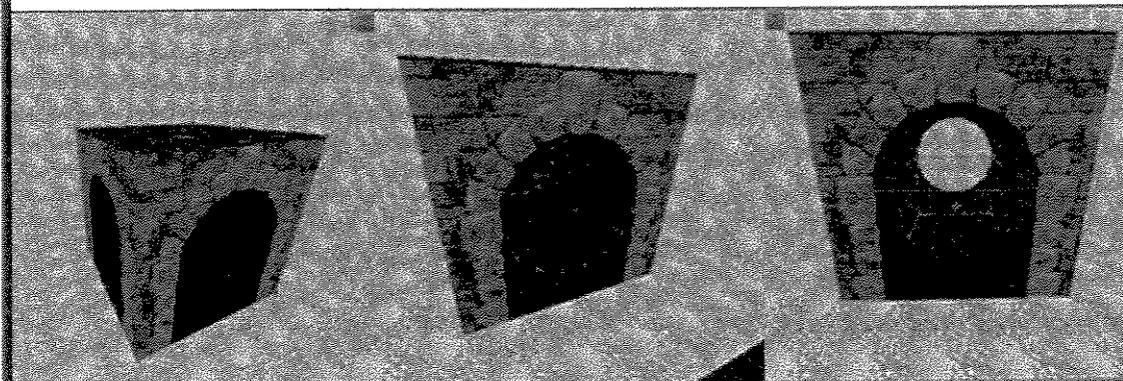
A construção de um Arco Românico, passo-a-passo usando o Unreal Ed



Constrói-se um cubo como base do arco.

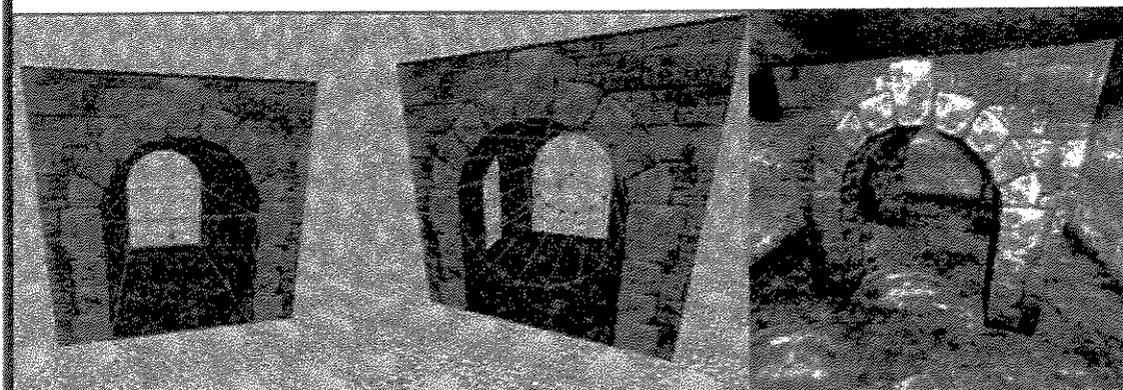
Escolhe-se a textura adequada.

Aplica-se a textura ao cubo.



Ajusta-se a textura no tamanho correto para cobrir o objeto.

A seguir, subtrai-se do cubo um cilindro e um paralelepípedo.



Repete-se esta operação para formar os demais vãos do arco.

O Arco pronto

Figura 2.6 - A construção de um Arco Românico, passo-a-passo usando o Unreal Ed

Texturas

Grande parte da qualidade visual de um ambiente virtual é atribuído a qualidade de suas texturas. São elas que realçam os detalhes dos objetos e nos convencem de que o que vemos não são simplesmente formas virtuais "sem vida". A textura reveste o objeto 3D como um papel de parede.

Na Figura 2.7 temos o exemplo de adequação de texturas diferentes em um mesmo objeto tridimensional. Em (a), o objeto é revestido com uma textura de rocha, ela tem tamanho reduzido e não apresenta nenhum detalhe que realce o objeto 3D. Em (b), temos a textura elaborada especialmente para revestir os arcos góticos. Existem detalhes do tipo de revestimento, detalhes de blocos de pedra na base e rachaduras.

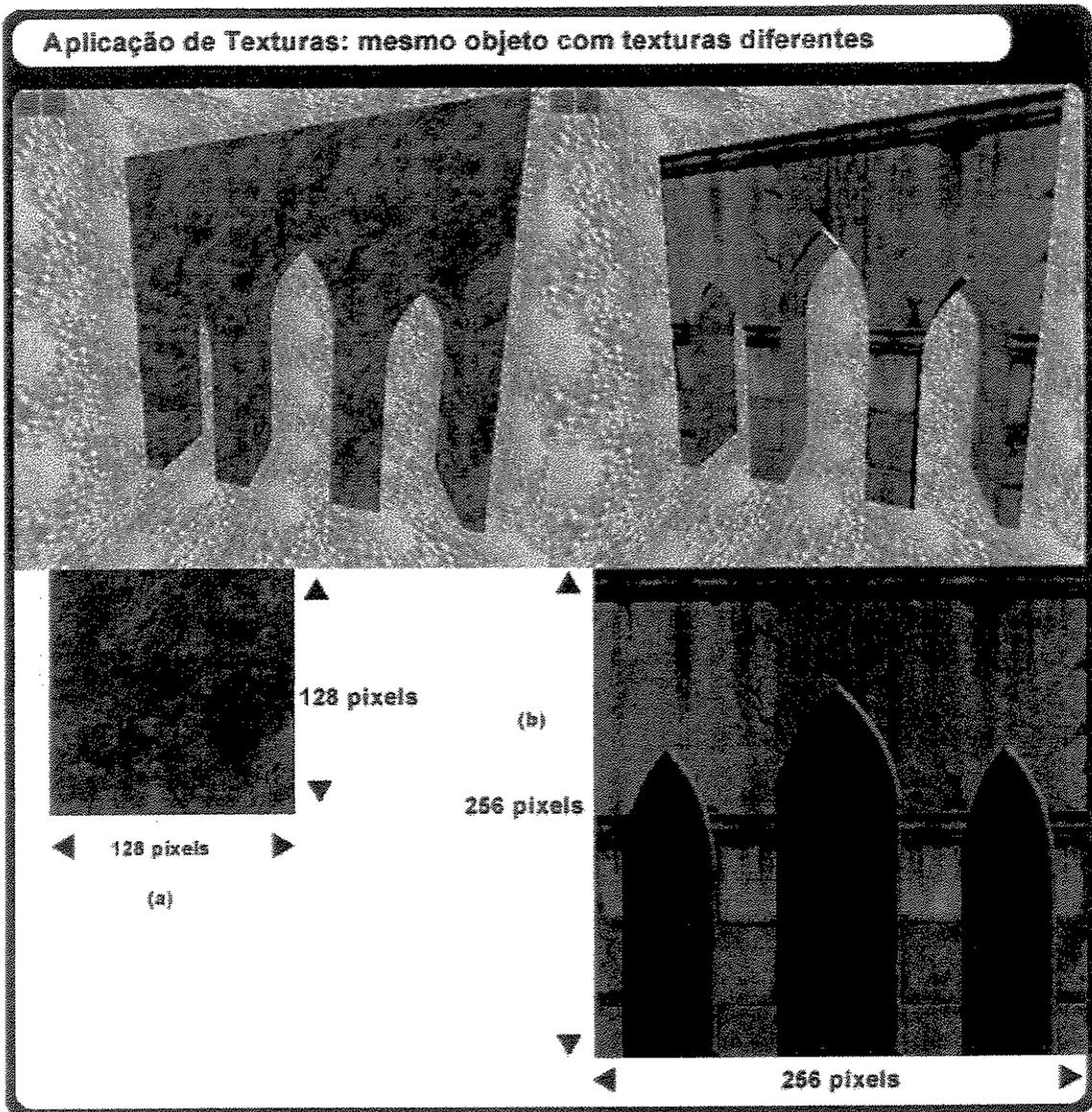


Figura 2.7 - Aplicação de Texturas: mesmo objeto com texturas diferentes

Como se Faz uma Textura

A base para a criação de uma textura pode ser uma foto, um desenho, enfim qualquer imagem digitalizada. A partir daí segue a edição um software de edição de imagens como o Adobe Photoshop, Gimp, Corel Photopaint, para citar alguns. Dependendo da complexidade da textura, o trabalho pode envolver o uso de múltiplas camadas, renderização de efeitos, etc. Mas, o usual é adequação deste material no formato que *engine* lê, principalmente nos seguintes pontos:

- **tamanho das imagens.** isto depende de cada *engine*, mas o padrão de tamanhos é sobre a potência de base 2. (32,64,128,512)
- **paleta de cores** (geralmente 256 cores). O uso de paletas de cores reduzidas, (fotos digitais usam uma paleta com no mínimo 16 milhões de cores) têm a função de não sobrecarregar o processamento e a memória durante a execução do jogo.

O Exemplo do Vitral do Profeta

Para caracterizar o visual gótico e religioso da fase era necessário a inserção de vitrais. As imagens foram digitalizadas em alta resolução (300dpi) das fotos dos vitrais da Catedral de Santa Maria em Augsburg na Alemanha.

As dimensões do arquivo de alta resolução não são incompatíveis com as necessidades de um jogo, cada vitral ficou com aproximadamente 1200x2700 px(pixel), ocupando 17Mb; se multiplicarmos por quatro (números de vitrais diferentes) chegamos a soma de 78Mb, portanto o primeiro passo é adequar suas dimensões e formatos.

A imagem foi reduzida para 256x640 px com mínimas distorções. A seguir foi dividida em dois outros arquivos: o topo com 128x256 px e a base com 256x512 px (mantendo as proporções do UnrealEd). A paleta de cores foi reduzida de 16 milhões para 256, os arquivos finais ficaram com foram exportadas no formato .pcx²⁰ ocupando em média 100kb.

²⁰ PCX é o formato usado comumente pelos computadores baseados na plataforma PC-IBM e compatíveis. A maioria dos programas para PC lêem a versão 5 do pcx que suporta paletas personalizadas.

Texturas: O Vitral do Profeta

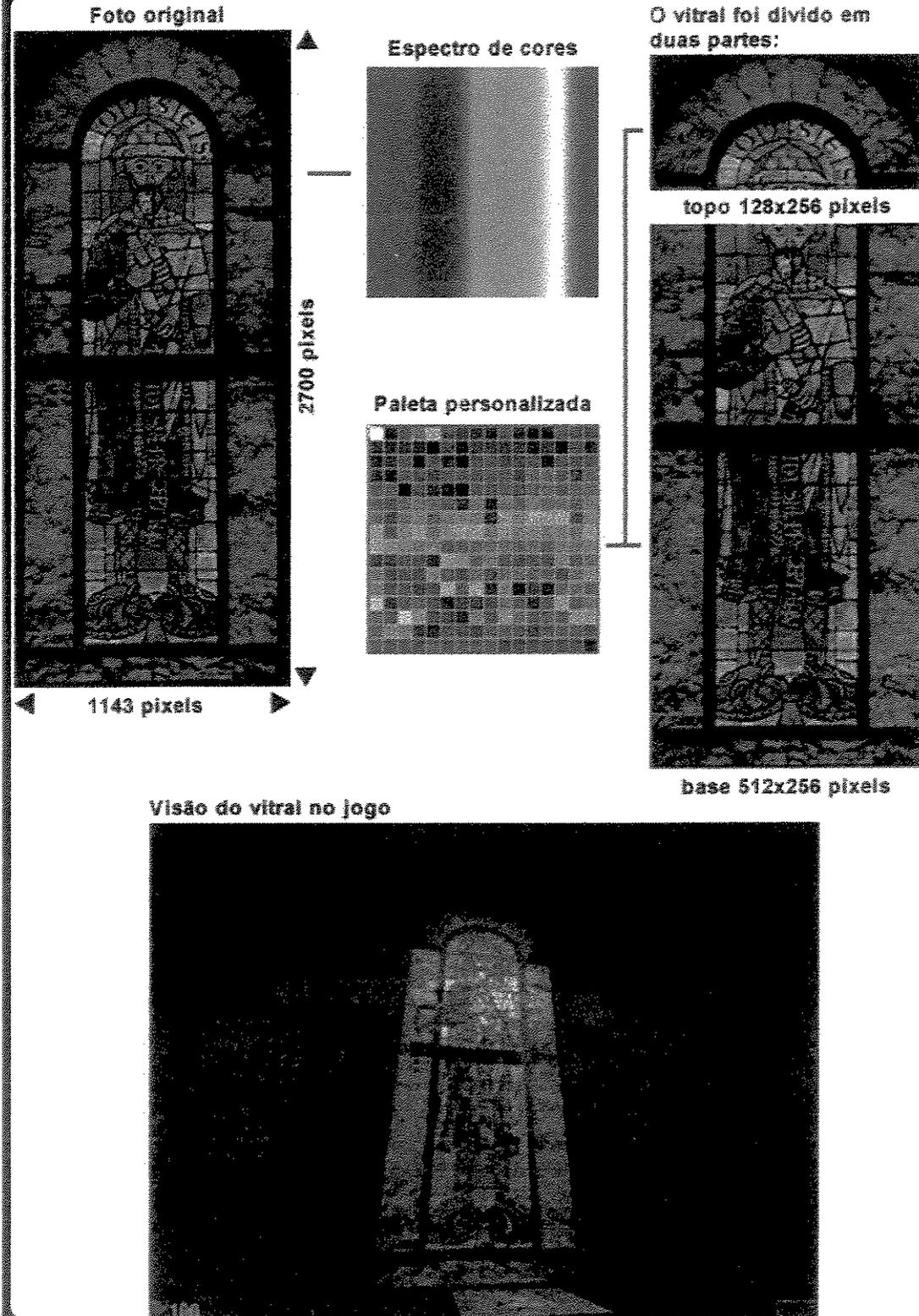


Figura 2.8 - Texturas: O Vitral do Profeta

Luz

A luz é um dos elementos mais importantes em ambientes virtuais. Mais que simplesmente iluminar, a luz é um elemento de controle poderoso. Dá ao designer, o poder de enfatizar o que deve ser visto e como deve ser mostrado.

Esta importância é pensada desde a programação. A luz nos editores de jogos é uma classe específica com vários parâmetros editáveis, em geral permitem a programação de scripts para funções especiais como, por exemplo, Trovões.

O UnrealEd usa o modelo HSL - Hue (matiz), Saturation (Saturação) e Luminance (Luminosidade/Luminância) para a determinação de cores, possui parâmetros como: Raio, Volume de brilho, Período, Fase, Brilho de Lente, etc. (ver fig. Propriedades da Luz no UnrealEd) A combinação destes parâmetros gera outros efeitos luminosos, um exemplo: uma luz do tipo *searchlight* (facho de luz direcional) com um período de rotação de 32 segundos, pode ser usado como a luz de farol marítimo.

A liberdade do designer é muito grande, a iluminação não necessariamente precisa ter relação direta com as fontes luminosas visíveis ao jogador, a luz de um ambiente não precisa vir exatamente da tocha que arde sob o teto. Elas podem estar distribuídas para realçar as sombras, ou iluminando um objeto específico que tem importância na cena²¹.

²¹ Num cenário gótico a luz tem uma função muito marcante. Os historiadores contam que a catedral gótica é o espelho da relação do homem medieval com Deus. A vida do homem ocidental era governada pelos dogmas da Igreja Católica em níveis bem mais profundos que os atuais. Questões que hoje podem ser classificadas de transcendentais faziam parte do dia a dia das pessoas. A noção de Purgatório não existia e o homem medieval vivia sobre a eterna dúvida da salvação de sua alma. A catedral gótica é uma metáfora do caminho do homem das trevas até a luz. Por isso sua grandiosidade, sua altura, a escuridão nos andares mais baixos e a luz sempre vinda do alto.

Propriedades da Luz no UnrealEd

Light Properties

- +Advanced
- +Collision
- +Display
- +Events
- +Filter
- LightColor

LightBrightness	64
LightHue	0
LightSaturation	255
- Lighting

bActorShadows	False
bCorona	False
bLensFlare	False
bSpecialLit	False
LightCone	128
LightEffect	LE_Searchlight
LightPeriod	0
LightPhase	0
LightRadius	64
LightType	LT_Steady
VolumeBrightness	128
VolumeFog	0
VolumeRadius	0

Algoritmos para efeitos de luz

- LE_None
- LE_TorchWaver
- LE_FireWaver
- LE_WateryShimmer
- LE_Searchlight
- LE_SlowWave
- LE_FastWave
- LE_CloudCast
- LE_StaticSpot
- LE_Shock
- LE_Disco
- LE_Warp
- LE_Spotlight
- LE_NonIncidence
- LE_Shell
- LE_OmniBumpMap
- LE_Interference
- LE_Cylinder
- LE_Rotor
- LE_Unused

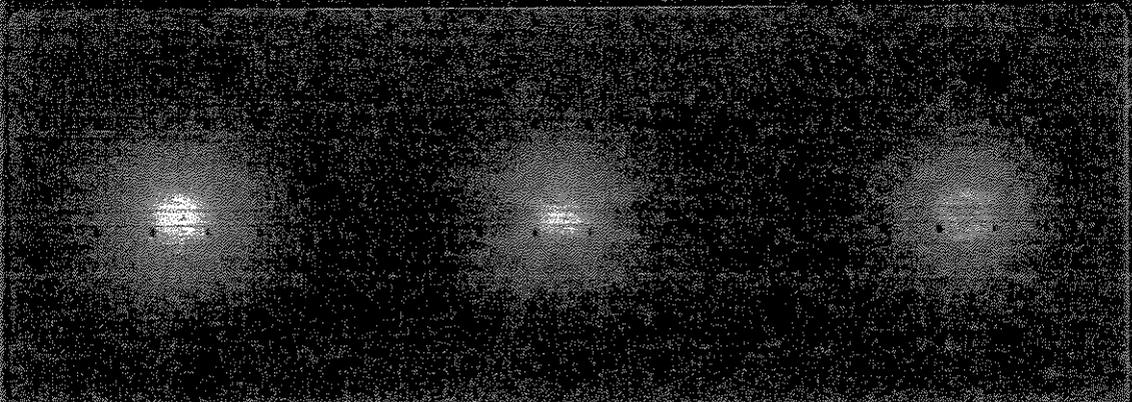
Tipos de Luz

- LT_None
- LT_Steady
- LT_Pulse
- LT_Blink
- LT_Flicker
- LT_Strobe
- LT_BackdropLight
- LT_SubtlePulse
- LT_TexturePaletteOnce
- LT_TexturePaletteLoop

O modelo de cores HSL —

Figura 2.9 - Propriedades da Luz no UnrealEd

Alguns Efeitos de Luz do UnrealEd



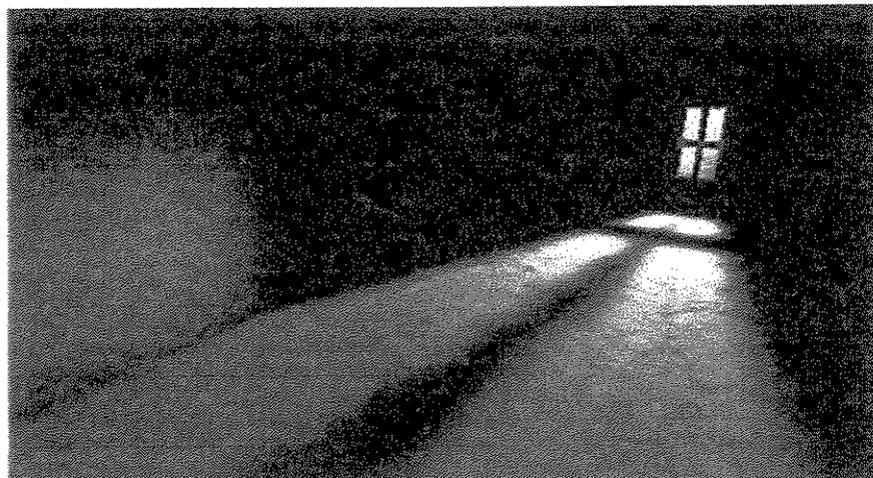
Luz padrão

Efeitos de ondas: Lenta e Rápida



Efeito Farol

Efeito Rotor



Combinação de luz, forma e textura com opção de sombras detalhadas (*High Shadow Detail*).

Figura 2.10 - Alguns Efeitos de Luz do UnrealEd

Catacumbas: imagem só com texturas e imagem do jogo pronto



Figura 2.11 - Catacumbas: imagem só com texturas e imagem do jogo pronto

A Luz do Vitral



Efeito de Brilho de Lente, simulando uma fonte externa de luz muito intensa.

A sombra da seta no chão indica qual caminho.



Ao caminhar por debaixo do vitral, o brilho desaparece.



O efeito das sombras em arcos é dado por a luz estar dentro de um cilindro.

A seta no chão é a sombra projetada de um cone. Ele também serve para bloquear a auréola em certos ângulos.

Figura 2.12 - A Luz do Vitral

Áudio

O Áudio nos jogos trabalha em várias camadas que se sobrepõem durante o ato de jogar (*gameplay*). O design estrutural de como estas várias camadas vão interagir é determinante para todo o desenvolvimento do resto do produto. Qual o número máximo de camadas podem ser executadas, quais têm prioridade, o quanto da capacidade do processador de memória são consumidos, quais os formatos usados, quais os vantagens/desvantagem (*trade-offs*) de cada um, resumidamente falando.

Cada projeto tem sua especificação, mas de forma geral pode-se dizer que as camadas principais são assim organizadas²²:

Camada 0 - Som Ambiente

O som ambiente é composto pelos vários sons associados a cada seção/região dentro do jogo. Exemplo: No cenário da Cripta Templária são os sons de água corrente, nas Catacumbas e na Torre são os diferentes tipos de ventos.

Camada 1 - Som Ambiente Incidental

É aquele disparado por um evento/ação do jogador, geralmente tem curta duração e enfatiza algum aspecto daquele evento. Exemplo: quando o jogador toca na estátua do monge, ouve um grito desesperado.

Camada 2 - Sons dos Personagens e Interface

Os personagens e elementos de cenários do jogo tem sons próprios que servem para caracterizá-los. Respiração, barulhos de animais máquinas, sons da interface, sons de aviso (ex.: insira o CD 2 para jogar).

Camada 3 - Música

É a camada que contém a Trilha Musical do jogo. Suas aplicações são discutidas no capítulo 3.

²² Em Zizza, Keith: *Your Audio Design Document: Important Items to Consider in Audio Design, Production, and Support. Gamasutra*. 26 de julho de 2000.

Camada 4 - Narração

É a camada dedicada às falas dos personagens e do narrador.

Camada 5 - Transmissão/Recepção de Voz

Esta camada é usada por jogos multijogador em rede que suportam tecnologia de voz sobre IP²³. Cada jogador com seu microfone pode falar com sua equipe, com todos os jogadores.

A Trilha Sonora de um jogo é a soma de todas estas camadas e deve ser para o jogador uma experiência prazerosa, que se integre coerentemente com os esforços da área gráfica, do design e dos outros elementos do jogo.

Os aspectos técnicos de áudio serão abordados no capítulo 3

²³ A voz captada no microfone é digitalizada e um programa específico que a transmite em forma de dados para outro ponto da rede onde é decodificada e convertida em sinal analógico.

Interação

O termo Interação em sua concepção mais ampla, mereceria um capítulo a parte. Para esta seção, o termo interação se refere somente à estrutura/sistema que monitora as ações do jogador (entrada de dados) e comanda as respostas dadas pelo jogo (saída de dados). O mecanismo pode ser descrito em linhas gerais:

- O jogador controla uma entidade (*player*) que faz o papel de “corpo” do jogador quando ele transita pelo ambiente virtual. Suas propriedades e movimentos são capturadas e interpretadas pelo *engine*. As mais importantes são: colisão²⁴ (*collision*) e posição espacial dentro do ambiente (coordenadas x,y,z).
- Para os jogos de ação em primeira pessoa o corpo do jogador tem um tamanho delimitado de colisão dado em altura e raio, medido em pixels.
- Outros eventos de entrada capturados pelo *engine* são: os tiros disparados pelo jogador, colisão de personagens controlados pelo *engine*, cliques de mouse e comandos de teclado.

Mover

O *mover*²⁵ é uma classe especial de objetos que tem liberdade total de movimentos dentro do ambiente virtual, independente da ordem de renderização. O *mover* não tem forma definida, para tal usa uma forma 3D pronta. O designer cria a forma desejada: porta, elevador, nave, etc. e após ter finalizado o objeto, insere-o na fase com a propriedade de *mover*.

O UnrealEd permite que este novo objeto tenha até oito posições diferentes no ambiente, bem como liberdade de movimentos de rotação nos três eixos (x,y,z). Após acionado pela colisão do jogador ou através de um *trigger*²⁶, o *mover* se desloca pelos pontos predeterminados obedecendo os parâmetros programados pelo designer.

²⁴ Colisão é o conceito que associa um volume virtual determinado a todos os objetos contidos em ambientes virtuais. Isto implica que esses objetos podem colidir, se bater, se tocar, numa metáfora da colisão física. O princípios de Colisão surgiram exatamente em *Spacewars* de Steve Russell.

²⁵ *Mover* é o nome usado pelo UnrealEd para determinar esta classe específica de objetos. Os programas baseados na tecnologia da Id Software usam a expressão “Brush-based entities” que em linhas gerais significa: objeto baseado em uma forma.

²⁶ *Trigger* (gatilho) é classe responsável pela captura da colisão do jogador, faz o papel dos sensores. O *trigger* também pode ser acionado por outro *trigger* ou ainda por *mover*.

A Fig. 2.13 ilustra como é feito um *mover* e seu funcionamento dentro da fase. O *mover* do exemplo é o elevador/plataforma que dá acesso aos porões da Catedral e se encontra no final da escadaria em caracol.

Em (a), vemos a plataforma em seu formato final. Ela foi obtida pela combinação de um paralelepípedo que faz o papel de grade e de uma seção de 1/8 de um octógono (um cilindro com oito faces) que serve como base. Em (b), as duas formas são unidas para dar origem a uma geometria única. As linhas em vermelho indicam as faces desse novo objeto. Após ser gravado, este objeto/geometria poderá ser usado em qualquer ambiente, bastando somente “importá-lo” deste novo arquivo. Em (c), o elevador está inserido no topo da escada em caracol com a propriedade de *mover*, note que ele é representado somente pelas linhas roxas. Em (d), vemos a primeira janela de propriedades do *mover*, a seção chamada “*Mover Keyframe*” é a responsável por guardar as posições do *mover* dentro da fase, note que dentro de suas opções existem oito “*Keys*” (Key 0 (base) ... Key 7), cada par de pontos seqüenciais define um segmento de reta. Quando o jogador aciona esse *mover*, o UnrealEd, faz com que o objeto percorra os pontos intermediários desse segmento de reta. Em (e) o elevador está em sua posição final. Em (f), temos o mesmo cenário com a iluminação final. Perceba que as outras geometria em roxo são os vários *movers* da fase da Catedral.

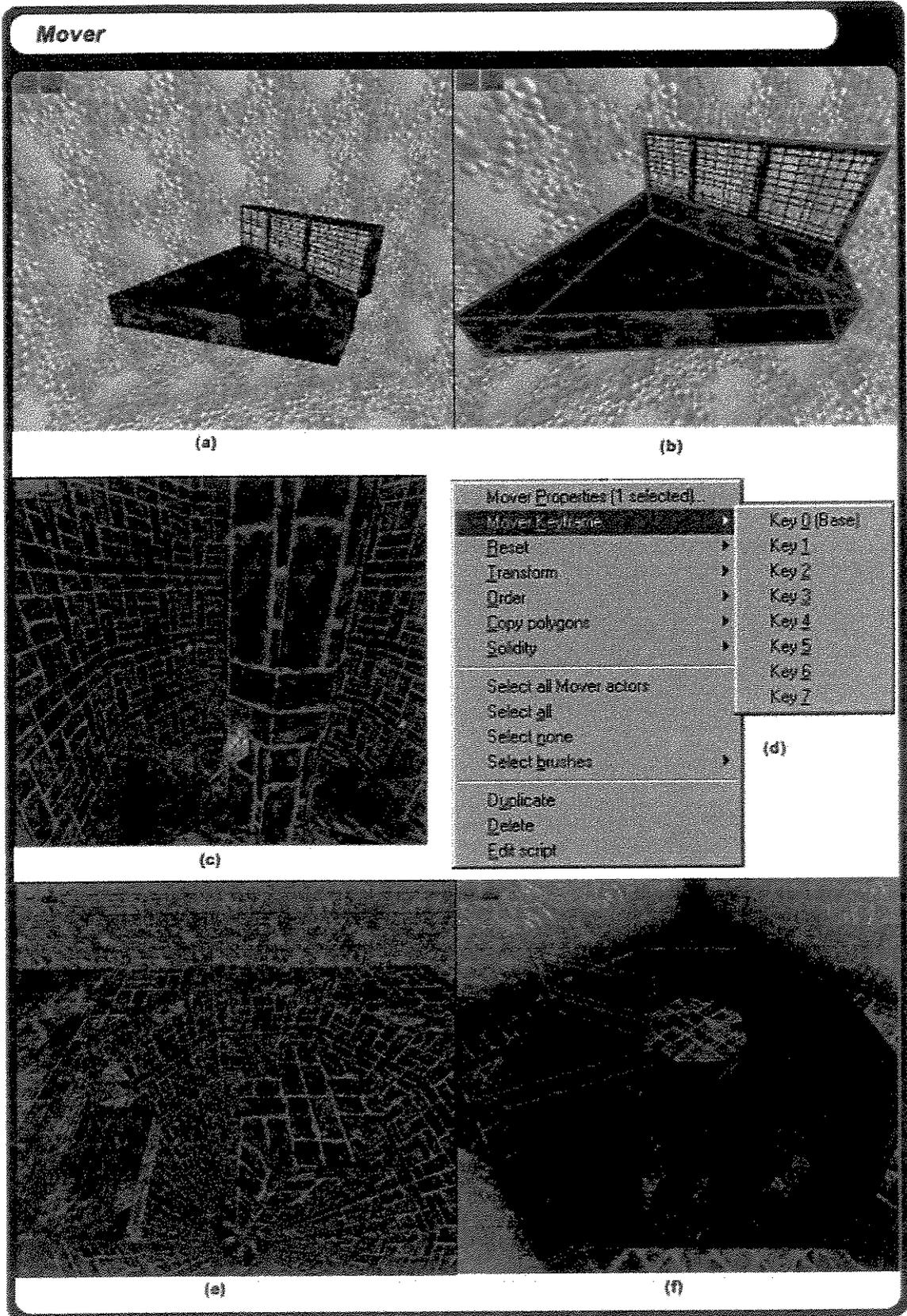


Figura 2.13 - Mover

O *mover* é a parte do cenário que reage aos atos do jogador e portanto é o indicador mais rápido de *feedback* do ambiente. Só uma arquitetura e texturas de qualidade não são suficientes para manter o jogador dentro do ambiente, ele precisa de respostas para saber se está no caminho certo, o resultado de seus atos tem de ser claros, mesmo que tal mecanismo de ação/resposta faça parte de um grupo de interações mais complexas.

Na Figura 2.14, temos um exemplo da *janela de propriedades* do *mover*. Em (a), temos a seção de entrada e saída de eventos/dados: *Event* define o evento de saída gerado pelo objeto(*mover*) quando acionado. *Tag* (identificador) define o evento de entrada que aciona o objeto(*mover*). Este evento de entrada é gerado por um outro objeto através de seu *Event*. Exemplo: O elevador está programado para captar o *Tag* “*Elevo*” que é gerado pela alavanca que fica na Capela Subterrânea (um outro *mover*) cujo *Event* é “*Elevo*”. Em (b), temos a seção de parâmetros de colisão e de movimentos. A opção *BT_PlayerBump* indica que este *mover* será acionado pela colisão de um jogador. Com relação ao movimento, a opção *NumKeys* indica que o *mover* tem duas posição definidas e a opção *MoveTime* indica que ele levará 5 segundos para percorrer a distância entre as duas. Em (c), vemos a seção de sons associados ao *mover*. O UnrealEd permite que o designer use até cinco sons diferentes para cada *mover*. *ClosedSound* – é som emitido quando o *mover* atinge a posição final (ex. batida do elevador na base da escada), *ClosingSound* – som que precede o fim do movimento (ex. o elevador “freando”), *MoveAmbientSound* – é o som em loop que acompanha o movimento do *mover* (Ex. som do elevador durante o trajeto), *OpenedSound* – é emitido quando o *mover* retorna a sua posição original, geralmente usa-se o mesmo o som do *ClosedSound*, *OpeningSound* - é som acionado logo após o início do movimento (ex. elevador “acelerando”).

As Propriedades de um Mover

Mover Properties

- +Advanced
- +Brush
- +Collision
- +Display
- Events

Event	None
Tag	Elevon
- +Filter
- +LightColor
- +Lighting
- +Movement

Mover	
bDamageTriggered	False
bDynamicLightMover	False
BrushRaytraceKey	0
bSlave	False
bTriggerOnceOnly	False
BumpEvent	None
BumpType	BT_PlayerBump
bUseTriggered	False
DamageThreshold	0.000000
EncroachDamage	0
KeyNum	0
MoverEncroachType	ME_IgnoreWhenEncroach
MoverGlideType	MV_GlideByTime
MoveTime	5.000000
NumKeys	2
OtherTime	0.000000
PlayerBumpEvent	None
ReturnGroup	None
StayOpenTime	4.000000
WorldRaytraceKey	0
- MoverSounds

ClosedSound	Sound'DoorsAnc.Other.chnd2e'
ClosingSound	Sound'CaithSound.Pilar.CchdLT'
MoveAmbientSound	Sound'CaithSound.Pilar.CsLT'
OpenedSound	Sound'DoorsAnc.Other.chnd2e'
OpeningSound	Sound'CaithSound.Pilar.CchdLT'
- +Networking
- +Object
- +Sound

Figura 2.14 - As Propriedades de um Mover

(a) seção para direcionamento de Eventos

(b) propriedades de Colisões e Movimentos

(c) opções de Sons

Triggers

Trigger é o “sensor” do ambiente, possui parâmetros programáveis de entrada e saída de dados, que podem ser direcionados para um outro *trigger* ou para qualquer outra classe programada para interpretar estes dados. O *trigger* pode ser configurado para captar somente um determinado tipo de evento como: presença do jogador, presença de personagens controlados, tiros, etc.

Geralmente serve para acionar portas, alavancas, elevadores, pontes, despertar “personagens adormecidos”, enfim mudar o estado (*status*) ou situação de objetos específicos dentro do jogo.

A Figura 2.15 mostra as propriedades de um *trigger*

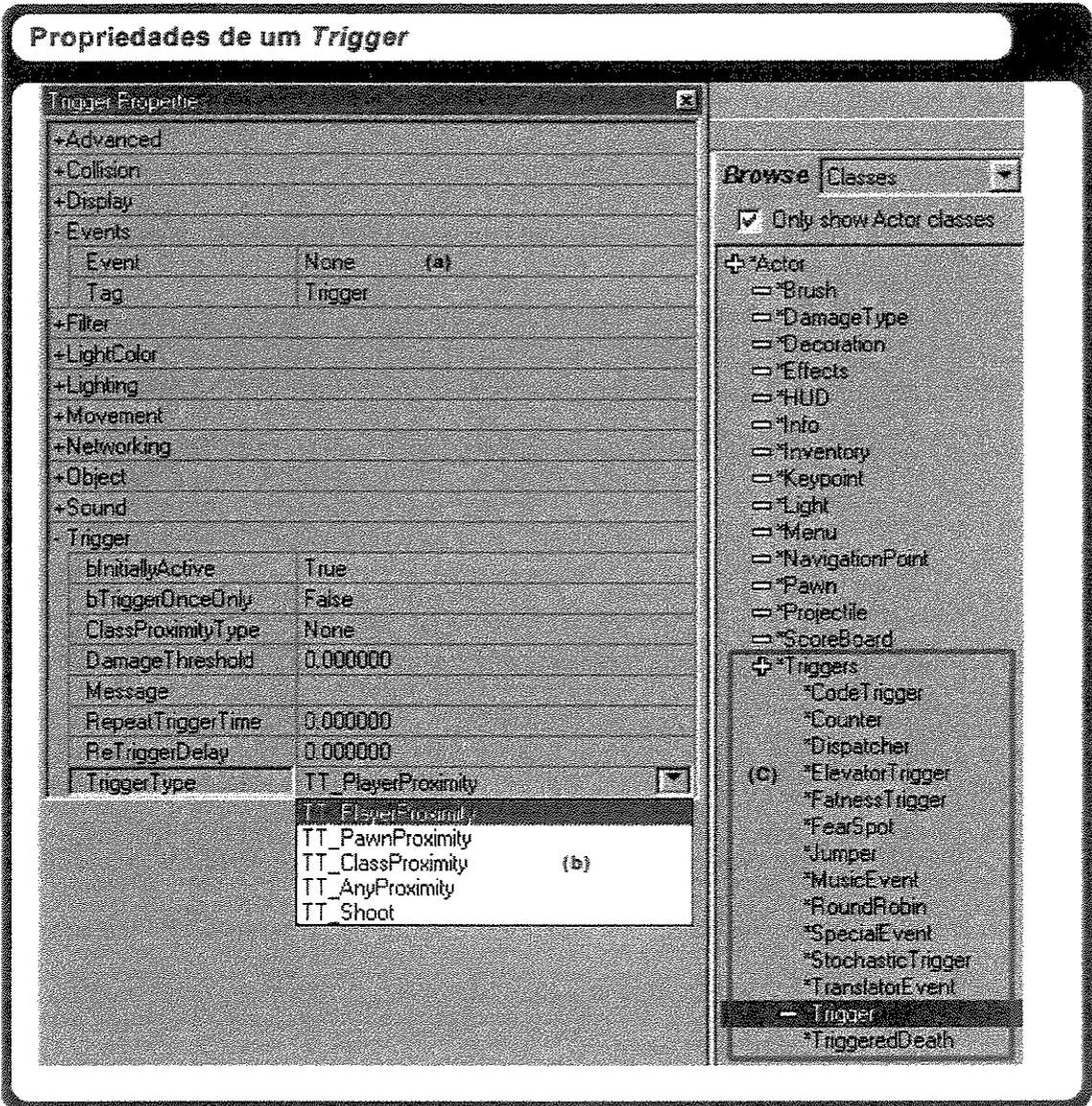


Figura 2.15 - Propriedades de um Trigger

Arranjos e Redes de Triggers

Os *triggers* podem ser arranjos de várias formas com outros triggers e também com outras classes para se obter formas de interações mais complexas. O arranjo dos eventos de entrada/saída permitem que o designer crie portas inteligentes, armadilhas, puzzles, etc. É a combinação de *movers* e *triggers* que dão vida e complexidade ao ambiente. Enfatizando a reação do ambiente aos atos do jogador, tornando atrativa a experiência de jogar.

A figura 2.16 mostra como pode ser montada um porta com dois arranjos diferentes entre *triggers* e *movers*. Em (a), temos o cenário composto por duas salas conectadas por uma porta. O ícone do *joystick* indica a posição de inicial do jogador e o paralelepípedo roxo é o *mover* em forma de porta.

Em (b), o arranjo não contém *triggers* para acionar a porta, o jogador terá de tocá-la para que se abra. O *mover* capta a colisão do jogador, se desloca para sua próxima posição (Key_1), permanece assim por poucos segundos e retorna ao seu estado inicial (Key_0). A consequência deste arranjo é que em uma das salas, quando o jogador tocar a porta ela abrirá em sua direção, captará sua colisão novamente e retornará para Key_0, entrando em ciclo vicioso.

Em (c), foram adicionados dois *triggers*, um em cada sala. Desta forma, o jogador conseguirá abrir a porta passando pela área de colisão de qualquer deles e a porta se abrirá independente de sua posição (jogador) dentro da sala.

Arranjos entre Triggers e Movers

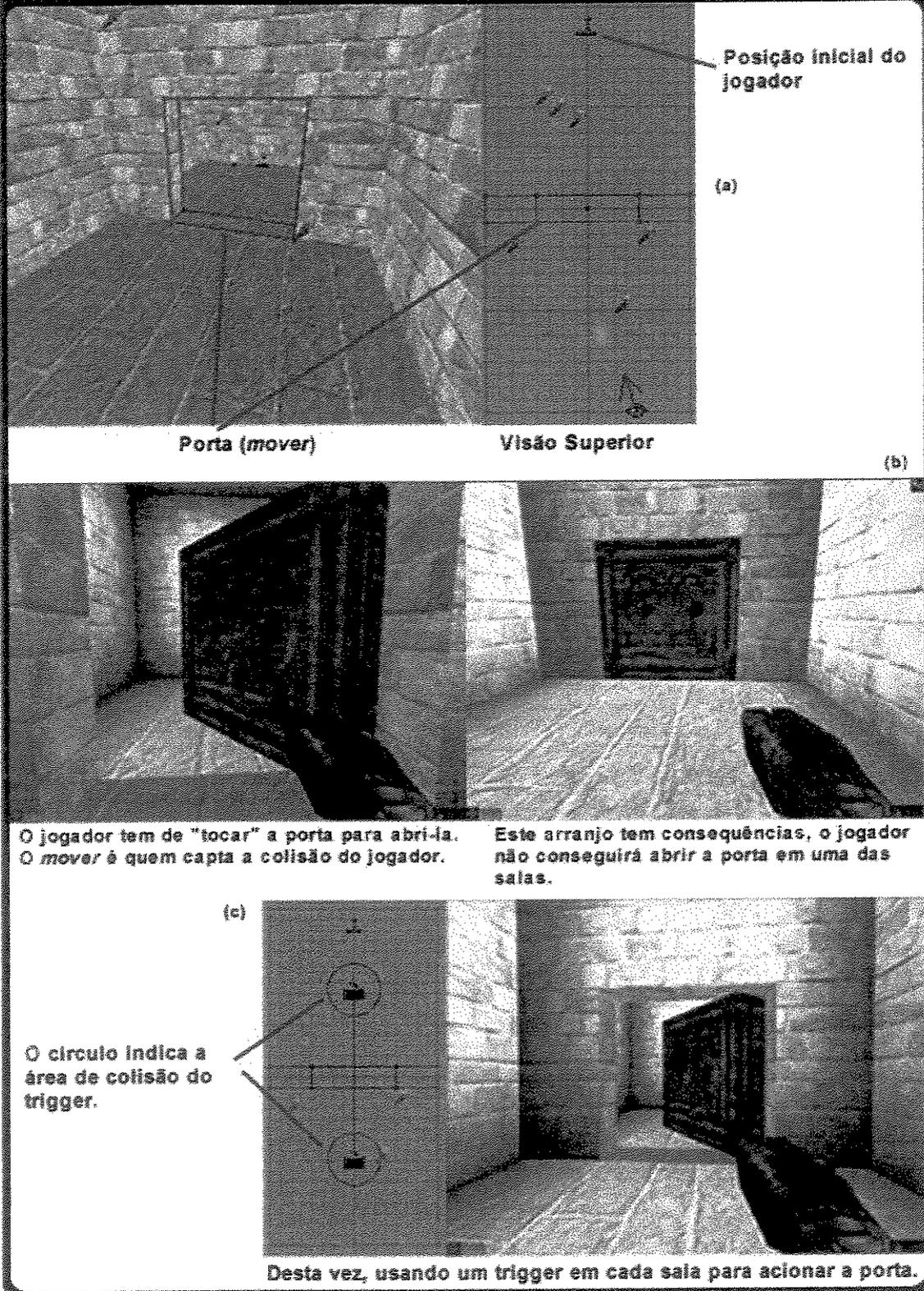


Figura 2.16 - Arranjos entre Triggers e Movers

O Pilar Escavado

A região do Pilar Escavado é a tentativa de se projetar uma cena onde o jogador possa encontrar a solução de um problema, caso se guie pelo design dos objetos e pelo *feedback* do ambiente.

A Figura 2.17 ilustra a situação passo a passo.

Em (a), o jogador vê um pilar diferente dos outros, ele tem duas de suas faces escavadas e por estes sulcos passam correntes. A intenção é informar o jogador de que este pilar tem uma função especial. Em (b), o jogador chega a um degrau/plataforma com grades que impedem sua passagem. Em (c), ele explora a região em volta e uma porta se abre dando acesso à capela subterrânea. Em (d), o jogador encontra a alavanca que aciona o elevador dentro da capela. A estrutura de *triggers* e *dispatcher* (trigger especial) envolvidos nesta cena é explicada na Figura 2.18.

Em (e), o jogador está de volta ao elevador que inicia sua subida e faz com que as lâminas do teto se abram. Em (f), para aumentar o "realismo" da cena foi adicionado, um grande cilindro é "iça" a plataforma.

A figura 2.18 mostra o arranjo de *triggers*, *dispatchers* e *movers* que compõe a cena, as imagens foram tiradas do UnrealEd. A seção (a) mostra o mapa orientado sob os eixos XZ (largura e profundidade) e (b) corresponde a orientação YZ (altura e profundidade). Os *movers* marcados são o elevador e a alavanca da capela. As linhas vermelhas do Unreal Ed indicam que estes *triggers*, *dispatchers* e *movers* estão conectados.

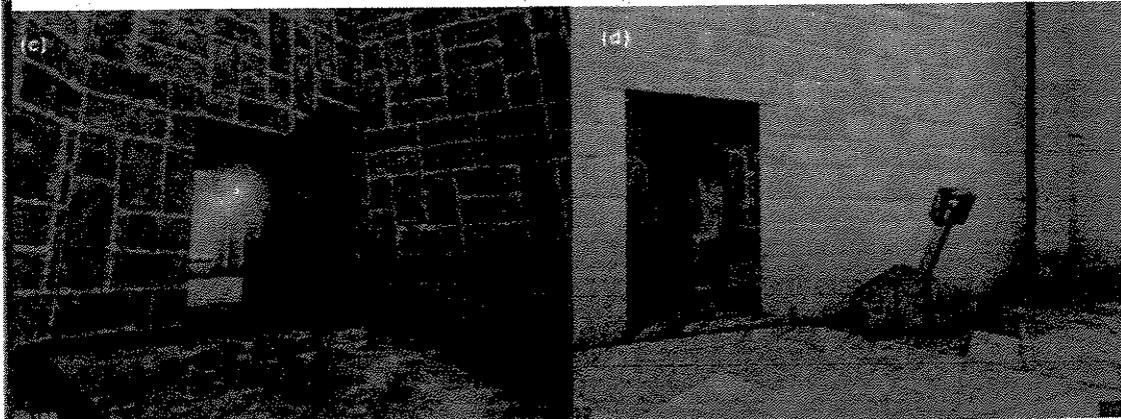
O mecanismo funciona da seguinte forma: o elevador está programado para receber somente a informação do *trigger* que ignora a colisão do jogador. Este *trigger* por sua vez, está desligado e só recebe informação da alavanca da capela. Quando o jogador empurra a alavanca ela envia a informação para ativar o *trigger*. O *trigger* liga o elevador que agora pode ativar o *dispatcher*. O *dispatcher* tem a propriedade de disparar até oito eventos diferentes temporizados. E por isso é usado para abrir as lâminas do teto uma a uma seqüencialmente.

O Pilar Escavado



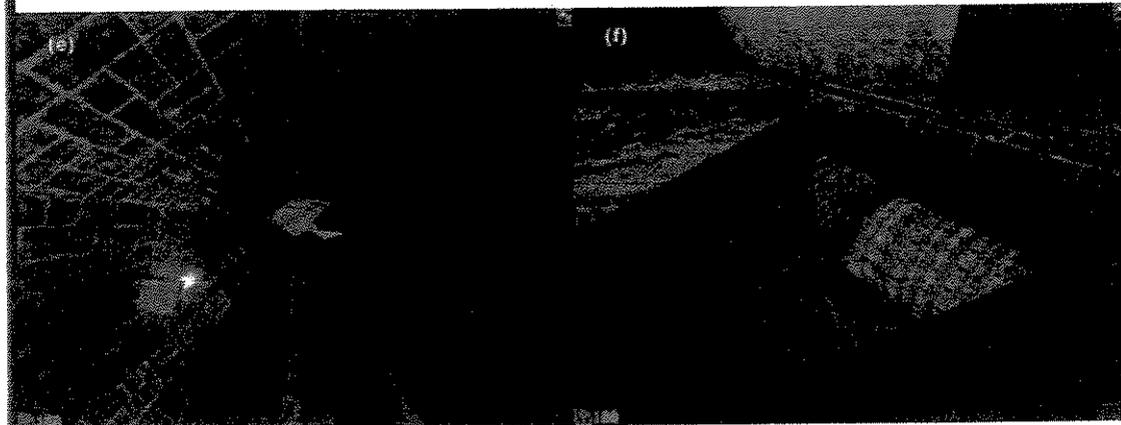
(a) Início da escadaria: pilar com correntes

(b) Fim da escadaria: plataforma com grades



(c) Ao explorar a região, a porta da Capela Templária se abre.

(d) A alavanca aciona o "elevador".



(e) As folhas do teto abrem seqüencialmente.

(f) O "mecanismo" iça a plataforma.

Figura 2.17 - O Pilar Escavado

Rede de Triggers

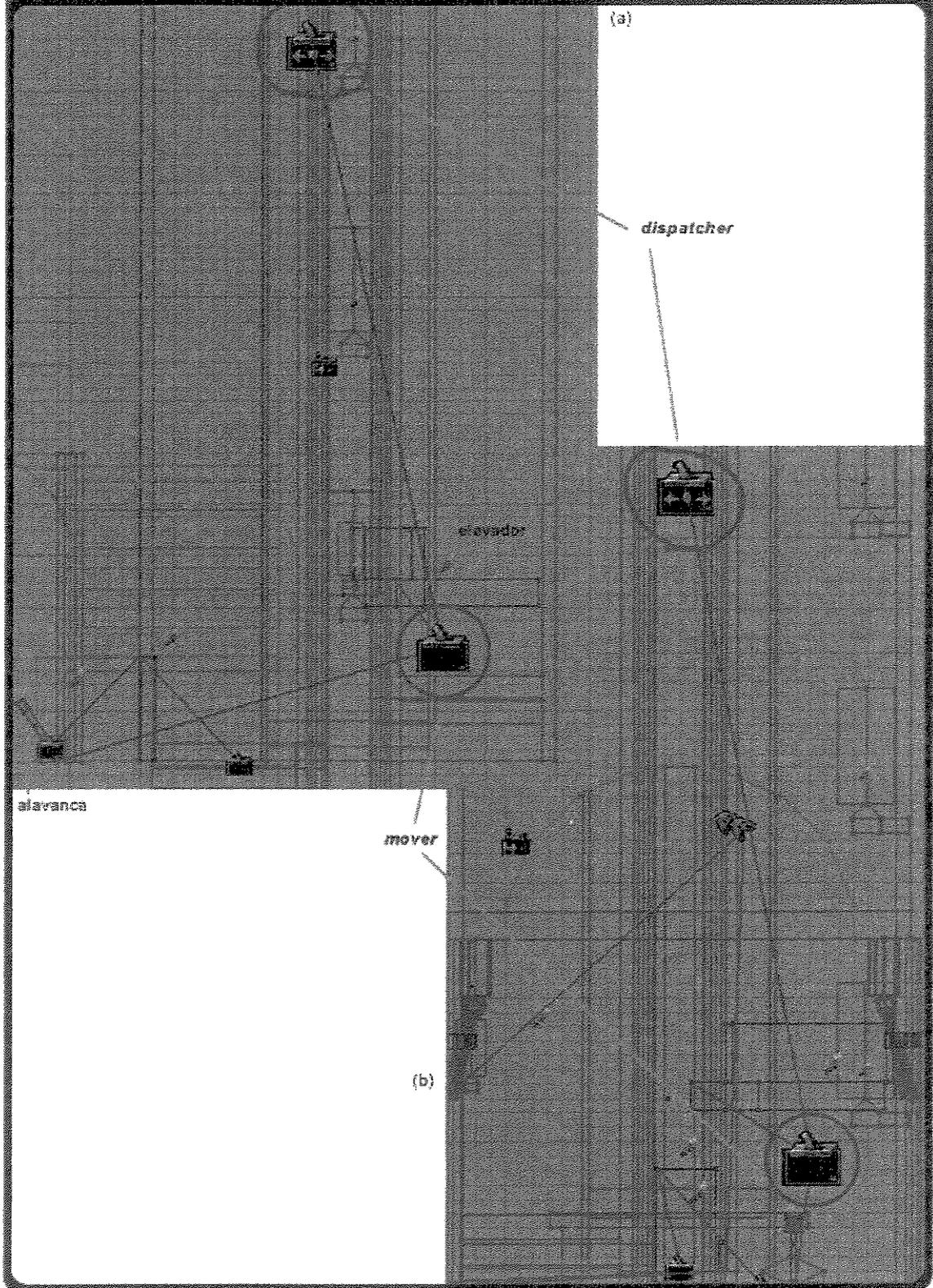


Figura 2.16 - Rede de Triggers

Capítulo 3 – A Música nos Jogos de Computadores

Design Sonoro

As imagens dos ambientes virtuais nos jogos são experimentadas aos poucos, sob um campo objetivo de visão delimitado pelo *engine*, o áudio pode transcender estas dimensões e envolver o jogador em uma atmosfera mais rica mostrando elementos e informações além da tela. O trabalho do designer sonoro é criar esta atmosfera, reforçando/alterando aspectos semânticos dos elementos apresentados no ambiente virtual como um todo.

Um modelo de organização do áudio nos jogos foi mostrada na seção **Áudio** do capítulo 2. O trabalho de caracterização do ambiente esta contido nas **Camadas 0 - Som Ambiente e Camada 1 - Som Ambiente Incidental** e a tecnologia para sua realização já está bastante maturada, comportando vários formatos de áudio em .wav. (ver quadro 3.1) em várias situações. Estas duas camadas podem formar uma camada maior que gostaria de chamar de **Efeitos Sonoros**

A característica principal da camada de Efeitos Sonoros é que todos os seus sons têm localização espacial definida dentro do ambiente virtual. Efeitos especiais relacionados a uma sala serão ouvidos somente naquela região, ou no máximo nas regiões próximas com volumes diferenciados.

As seguir nos concentraremos na música para os jogos pois é o foco deste trabalho.

Música para Jogos

Por muito tempo a música nos jogos eletrônicos ficou limitada aos sons básico como “beeps”, “poings”, ou melodias de duas notas. Mesmo quando estes padrões foram superados tecnologicamente, muito trabalho precisava ser feito até que meros sons chegassem a forma da música tal como a conhecemos. Por muito tempo, os produtores de jogos não consideraram a música como um grande diferencial de qualidade em seus produtos, a trilha musical não justificava grandes investimentos para a sua confecção.

Um sintoma grave mostrava o quanto a música dos jogos precisava mudar. Ao iniciar uma nova seção de jogo, a primeira ação de legiões de jogadores no mundo inteiro era: desligar a música!

Seria a música totalmente dispensável nos jogos eletrônicos? Nós achamos que não!

Até hoje não existe um trabalho teórico que aborde conceitos mais amplos sobre sua utilização da trilha sonora em jogos de computadores. Isto demonstra o quanto esta área foi negligenciada durante o período de consolidação da indústria de jogos. A despeito da pouca preocupação da dimensão musical dos jogos, a música abre um novo leque de possibilidades significativas nos jogos de computadores que ainda não foi explorada. São várias as questões que ainda não foram resolvidas. E as possibilidades de sua aplicação são muito maiores do que as atuais.

Sobre Não-linearidade dos Jogos

Um dos problemas mais imediatos a ser resolvido é a adequação da trilha sonora à não-linearidade inata ao jogo de computador. Neste aspecto o jogo se distingue das mídias lineares como o cinema, a televisão e o vídeo, nos quais a música flui linearmente. E as soluções para este problema ainda são muito limitadas.

Apesar de todos os aspectos não-lineares de um jogo de computador, a experiência do jogar sempre se dará linearmente, para o jogador o tempo sempre está correndo em seu sentido normal e esta característica, comum a qualquer mídia interativa digital é que permite o emprego da música em várias situações. A música possui uma característica de desenvolver-se no tempo, ou seja ela é em princípio linear. Numa situação como a encontrada nos jogos é preciso desenvolver novas soluções, novas técnicas, novos procedimentos que permitam adequá-la a esta estrutura não-linear.

Ao longo do tempo, os designers de jogos têm utilizado algumas soluções que se tornaram padrões tais como as estruturas cíclicas, uma das técnicas de composições minimalistas. Esse procedimento permite que a música seja cortada ou mudada durante a ação do jogador sem provocar uma interrupção muito brusca, sem desviar a atenção do jogador do jogo. Normalmente as transições entre um fragmento musical e outro estão

diretamente associados ao “clique” do usuário. Assim forja-se artificialmente uma situação de linearidade, onde em princípio ela não existe.

Uma Classificação Possível

Esta classificação do uso da trilha sonora, partiu da análise de jogos comerciais de 1994 até 2002. Vários gêneros foram analisados: Ação em Primeira Pessoa (First Person Shooter), Multijogador (Multiplayer), RPG (Role Playing Game), “Filmes Interativos”, Estratégia em Tempo Real (Real time Strategy).

- **Trilha de Ambiência:** tem a função de preencher o vazio sonoro. É composta de vários temas musicais que são usados aleatoriamente durante a ação, não influenciando no significado da mesma. Sua função é simplesmente a de fundo musical: pode ser a música vinda do “rádio” ou de alguma outra fonte que apareça durante a cena. Exemplo é o jogo “The Need For Speed - edição especial – Electronic Arts”, onde o jogador escolhe entre dois estilos musicais diferentes: Tecno e Rock; a partir daí, todas as suas corridas de carro serão acompanhadas por oito temas nesse estilo, apresentadas aleatoriamente.
- **Trilha Síncrona:** é usada quando o jogador não pode interferir na cronologia da ação. Independentemente do desempenho do jogador, os eventos do jogo ocorrerão a intervalos de tempo pré-determinados. Exemplo: O jogo “Rebel Assault II – Lucas Arts” nos traz uma aventura baseada nos personagens da série Guerra nas Estrelas. O jogo é basicamente um filme interativo onde o jogador tem de vencer fases e, após vencer cada uma delas, assistirá a uma nova parte do enredo. A trilha sonora é a mesma composta por John Williams para a série de cinema original. Para criar a sensação de ação intensa, os produtores do jogo fizeram com que as fases tivessem todos os seus momentos críticos coincidentes com os momentos críticos acentuados por John Williams em sua trilha. Com uma utilização

moderna do *mickeymousing*²⁷, o jogo realmente passa a sensação de que o jogador está “dentro do filme”²⁸.

- **Trilha por Fase:** cada fase tem seu tema musical e orquestração próprios que se sucedem conforme o jogador avança no jogo. A música faz a ambientação de cada fase, também fornecendo “atmosferas”, “intenções” e o ritmo da ação. Ao mudar de fase, o jogador terá nova música, nova ambientação e novo ritmo. Este tipo de trilha não tem o compromisso de enfatizar diretamente nenhum evento específico dentro da fase, mas sim o seu todo.
- **Trilha por Evento:** ocorre quando o “click” do usuário ativa imediatamente um tema musical curto, geralmente associado a um evento específico da ação. Esse tema se sobrepõe à música principal temporariamente, enfatizando algum aspecto daquele evento. Ele pode se relacionar a algum item do “inventário” do jogador (objetos virtuais coletados durante o jogo), ou ao uso de uma “arma especial”. Tem como objetivo sublinhar tal ação, dando importância àquele momento. Seu uso é mais comum nos jogos do gênero RPG, onde a interação do jogador depende mais de seu raciocínio e criatividade na soluções de “casos e mistérios”.

Em nosso modo de ver, tal classificação não esgota os usos possíveis da música nos jogos. A criação de ambientação em seu sentido mais amplo, por exemplo: tal como usada pela indústria cinematográfica, ainda não foi explorada em profundidade, alcançando o grau de sofisticação que a música atingiu no cinema, mas sem deixar de lado as especificidades e a necessidades dos jogos eletrônicos.

²⁷ O *mickeymousing* foi assim denominado na década de 30, para descrever o tipo de música característica dos filmes deste período, nos quais a trilha sonora lembrava a técnica de sonorização usada pelos desenhos animados da época. Sua principal característica é a de estabelecer um paralelismo rítmico entre o movimento visual e o sonoro.

²⁸ Até 1999, a Lucas Arts ainda utilizava uma técnica semelhante de *mickeymousing* como no caso do jogo: “Star Wars Episode I - The Phantom Menace”

As Tecnologias de Áudio e Design Sonoro dos Jogos

A Era MIDI

A primeira opção viável para a criação de trilha sonora musical mais elaborada com polifonia²⁹ surgiu em 1989 com a popularização das Placas de Som SoundBlaster™ da Creative Labs. Não era necessário mais ficar preso aos “beeps” e “poings” do Alto-falante do *Personal Computer*³⁰. As placas de som possibilitam entre outras coisas, o uso de arquivos no formato genérico: MIDI³¹.

O formato *.mid* pode ser comparado a um arquivo de texto: contém apenas as informações sobre a seqüência de notas musicais e algumas de suas características como: duração, intensidade, qual o “instrumento”, entre outros. Um arquivo *.mid* não possui amostras de áudio encapsuladas internamente, estas estão na memória da placa de som do usuário e são “disparadas” pelas informações do *.mid*.

Este tipo de arquitetura implica em algumas características muito importantes para a criação de música para jogos. O arquivo final com a trilha musical é pequeno, consome pouca memória e pouco processamento para ser executado. A placa de som é quem se ocupa em *renderizar* o áudio, liberando a CPU para processar os gráficos. O *.mid* não garante a mesma qualidade de áudio para todos os computadores, isto depende somente da qualidade da placa de som do usuário.

As limitações do *.mid* são muitas, bem o sabe quem ouviu a trilha sonora de *Star Wars* em *.mid*, a experiência é no mínimo estranha. As principais deficiências do *.mid* são quanto a performance “musical” e qualidade do áudio, Existem placas de som profissionais que suprem muitos destes “defeitos”, mas não representam um número

²⁹ Polifonia é um termo técnico em música que indica que várias vozes (sons) ocorrem em simultaneidade.

³⁰ O Alto-falante do computador (*pc speaker*) é um dispositivo de baixa resolução e não permite a execução de sons complexos.

³¹ MIDI – Musical Instrument Digital Interface. Criado pela Roland em 1983, o MIDI é um dos protocolos de padronização de maior sucesso na indústria musical. Ele permite que instrumentos musicais, samplers, sintetizadores, computadores, etc. possam “conversar” através de um protocolo comum. Apesar ter sido projetado para o padrão de 8 bits, seus conceitos de operação podem ser estendidos para qualquer plataforma que venha a ser criada com padrões mais modernos.

significativo de usuários. Nenhum desenvolvedor de jogos pensaria em criar uma trilha musical para ser ouvida somente em um sintetizador profissional de alta qualidade.

De 1989 a 1994, o que predomina nos jogos do mercado são trilhas musicas em MIDI. O máximo em termos estéticos alcançado neste período é a criação de um fundo musical para cada fase, como nos jogos **Wolfenstein**, **Doom**, **Doom II**, com sons não muito definidos e com muita repetição. **Duke Nuken**, de 1996, é um dos primeiros jogos a mudar um pouco esta abordagem e chega a ter um tema de abertura que faz sucesso entre os fãs e de certo modo “caracteriza” o personagem principal do jogo.

Áudio com Reamostragem

Em 1995, surgem as primeiras trilhas sonoras com qualidade de áudio próxima à qualidade do CD. Isto era feito com arquivos de áudio com taxa de amostragem baixa (arquivos pequenos), a idéia era fazer com que uma amostra com qualidade de 8 bits e 11 KHz soe como uma amostra de 16 bits³² em 22.1 KHz. O jogo “lia” os arquivos de áudio de 8bits e re-amostrava este arquivo conforme a escolha do usuário, em 11 KHz bits ou 22.1 KHz bits. Este tipo de abordagem exigia máquinas mais poderosas e o processador padrão para tal era o Pentium MMX (166 MHz -).

Os jogos com esse tipo de trilha são, por exemplo, os da série *The Need For Speed* (Eletronic Arts) e *Rebel Assault II* (Lucas Arts). A diferença do uso da música pelos dois estúdios, ilustra bem a diferença de formação dessas empresas. A Eletronic Arts, formada por designers saídos da Atari e da Intellivision, fez uma trilha sonora muito bem direcionada ao estilo do jogo, “canções” instrumentais que acompanham e fazem fundo para as corridas alucinantes de *The Need For Speed*, o resultado é uma trilha musical que sustenta e sugere o ritmo jogo, apesar de não ter compromisso nenhum com o sincronismo da ação³³.

³² São dois os fatores que influem na qualidade do áudio digital – **Taxa de amostragem** (sample rate) e **Profundidade de bits** (bit-depth). A taxa de amostragem indica o número amostragens feitas em um segundo e bit-depth (profundidade de bit) é a quantidade de bits (zeros e uns) usada para armazenar tal informação. Quanto maior a taxa de amostragem, mais fiel o sinal digital é do sinal amostrado e quanto maior a Profundidade de bits, com mais qualidade estes dados são descritos.

³³ Sincronismo: Este tópico será abordado com mais profundidade nas páginas seguintes, mas de antemão deve-se esclarecer alguns fatores que norteiam o que chama-mos de *Trilha Musical*. Nenhum desses fatores é mais contundente do que o Sincronismo entre a imagens em movimento e o som para que desta combinação surjam significados diferentes que se cada uma fosse vista separadamente.

Em **Rebel Assault II** da LucasArts, empresa do cineasta George Lucas, percebe-se uma ênfase muito pronunciada na intenção de sincronizar a ação na tela com a música da Série Guerra nas Estrelas. É a música quem dita o tempo da ação, num exagero, pode-se afirmar que o jogador é mais um passageiro do que um agente no jogo.

A Trilha Sonora com CD-Áudio, MP3 e Module File

Apesar da trilha sonora em CD no formato Redbook³⁴ ser aquela com a maior qualidade dos formatos atuais (excluindo DVD) é a que oferece menor possibilidade de uso, sua única aplicação é a de trilha de preenchimento. O controle sobre a mídia é mínimo, qualquer corte ou “mudança” de faixa fica muito evidente, comprometendo a fluidez da ação do jogador. Mesmo assim, alguns títulos merecem destaque como: **Quake** e **Quake II** (Id Software, Activision) com a trilha de Trent Reznor do Nine Inch Nails³⁵, numa tentativa não muito bem sucedida de unir a estética futurista da música com o jogo, a impressão que fica é a de se estar ouvindo um CD diverso do jogo.

Redneck Rampage (Xatrix Entertainment, Interplay) é um dos jogos de ação mais engraçados da última década, tem como cenário a cidade fictícia de Hickston no Arkansas, onde alienígenas dizimaram a população local e os substituíram por clones, mas devido à “grande visão” dos invasores, fizeram apenas dois modelos de humanos: um velhinho careca e um Caipirão de Jardineira Jeans. O jogo convoca o mote “sangue e bebedeira” e a seleção de bandas de *Southern Rock* e *Country* convida o jogador a participar do clima country-western do jogo. Mesmo mudando o suporte da trilha sonora, a forma de seu uso é igual a que se fazia em *The Need For Speed*.

A preferência dos desenvolvedores até agora, tem sido por formatos que possam ser armazenados no disco rígido do usuário e ocupem menos espaço como o: *.mp3* e os *Module files* (usado pelo Unreal Engine). O padrão de áudio digital para as plataformas PC é o formato *.wav* (wave). Seu uso é mais comum e na forma descompactada. O *.wav*

³⁴ O Formato RedBook é a norma do consórcio Phillips e Sony, criadores do formato CD Áudio. O Termo RedBook se refere à capa do livro que contém toda a definição técnica do formato CD.

³⁵ A banda americana foi muito conhecida por ser um dos ícones da estética cyber-seattle-trash-metal, mas os exemplos de Quake e Quake II mostram que existe uma grande diferença entre uma Trilha Musical feita por um trilhista profissional e a Trilha Musical feita por um músico de uma banda.

aceita várias taxas de amostragens diferentes e várias profundidades de bits conforme o Quadro 3.1

O **MP3** é o formato de compactação de áudio que reduz em até 12 vezes o tamanho original dos arquivos, baseado no tipo conteúdo. O algoritmo de compactação analisa arquivo de áudio e tenta eliminar o máximo de redundância dessa amostra.

O **Module file**³⁶ é um tipo de arquivo que encapsula várias amostras de áudio (ex. instrumentos musicais) e sua seqüência de execução. É como se fosse um *.mid* com o áudio embutido. A reamostragem do arquivo pronto é adaptado conforme a capacidade da máquina do usuário. Da mesma forma que o MP3, permite o controle rápido de acesso, possibilita o uso de transições de volume durante sua execução dentro do jogo, e permite que equipes de trabalho diferentes possam trocar estes arquivos via Internet.

³⁶ O *Module File* (.mod) é formato de música digital criado para o Commodore AMIGA e é a base para todos os outros formatos(S3m, IT e XM).

Quadro 3.1 – Quadro comparativo de Formatos de Áudio Digital

Tipo de arquivo	Taxa de amostragem (sample rate)	Profundidade de Bits (bit depth)	Tamanho de 1 min. de música estéreo
.MID	Não tem	Não tem	Aprox. 10 Kb
.IT, .S3M, .MOD, .XM	Depende das amostras	Depende das amostras	Entre: 500 Kb e 5 Mb
. WAV	11 KHz	8 bits	Aprox. 2 Mb
	22 KHz	16 bits	Aprox. 5 Mb
CD	44 KHz	16 bits	Aprox. 10 Mb
MP3	44 KHz	16 bits (taxa de transferência 128 Kbps)	Aprox. 1 Mb

A Trilha Sonora do Jogo Black Death

Para compor a trilha sonora de qualquer o jogo o designer só precisa de um bom editor de áudio, existem várias opções no mercado. Os arquivos só serão inseridos no jogo depois de finalizados e devidamente formatados.

O UnrealEd só trabalha com arquivos .wav com resolução máxima de: 22Khz e 16Bits. Os arquivos .wav são usados para efeitos sonoros, som ambiente (vento, chuva, etc.), falas e disparos de armas. Em geral, estes arquivos têm curta duração e dependendo da situação podem ser usados em loop.

O .wav é formato de áudio por excelência usado por qualquer profissional de música, televisão, vídeo e cinema. Mas, a maior limitação em se usar arquivos .wav para música no UnrealEd é que sempre este som terá uma localização espacial definida dentro do ambiente virtual. A classe que comporta o arquivo de áudio em .wav tem necessariamente uma posição espacial fixa. Isto implica que quando o jogador estiver explorando o ambiente, poderá distinguir a fonte sonora, pois o programa calcula em tempo real os efeitos de distância, reverberações, *Efeito Doppler* entre outros.

Estas propriedades são fantásticas e no jogo permitem que se obtenha várias “sonoridades” a partir de um mesmo arquivo de áudio. Este efeito, da fonte sonora explicitada dentro da cena, é largamente usado no cinema, pode ser observado em cenas onde a música é ouvida quando o personagem ou a câmera chega perto de um rádio contido no cenário. Mas, a articulação da música nos meios audiovisuais não se restringe somente a esta modalidade. A trilha musical no cinema, na maioria das vezes, não ocorre desta forma. Ela é ouvida sem que seja necessário justificar sua fonte, o exemplo mais comum deste tipo de articulação é encontrado nos filmes de ação³⁷.

O formato nativo do UnrealEd para música é o: .umx (unreal module extension). Ele pode ser gerado a partir de um arquivo do tipo *Module File* (.it, .s3m, etc.). Diferentemente dos arquivos em .wav comuns, o .umx é executado independente da posição do jogador dentro fase. Portanto, para usar música no UnrealEd, o *sound designer* deve criar sua música em programas do tipo *Module Tracker* como: o Modplug

³⁷ Estes conceitos de articulação da trilha musical no cinema podem ser vistos em: CARRASCO, Claudiney R. Trilha Musical: Música e Articulação Fílmica. Tese de Mestrado ECA/USP. São Paulo, 1993.

Tracker. E depois deve importar este arquivo via UnrealEd, que faz a conversão para .umx.

O ModPlug Tracker

O Modplug Tracker é um dos poucos editores de *Module Files* disponíveis que trabalha com o sistema operacional *Windows*. Este tipo de programas é indicado principalmente para a criação de músicas eletrônicas e se adapta muito bem essa estética. Favorece loops de instrumentos, melodia curtas, uso de instrumentos sintetizados, transposições, enfim trabalha muito bem com amostras de áudio pequenas. (até 100kB)

Funcionamento

O Modplug Tracker funciona como um sequenciador, com ele o usuário informa qual a nota musical deve ser tocada, qual instrumentos, qual o tempo de duração da nota, compasso, posição estéreo, para citar alguns parâmetros. Estas informações podem ser inseridas através do teclado do computador, ou através de instrumentos MIDI (teclados musicais e controladores). A diferença essencial entre um arquivo .mid e o *Module file* é que no .mid as amostras de áudio são lidas da memória ROM da placa de som, enquanto no *Module file* as amostras estão contidas no próprio arquivo da música. O Quadro 3.1 apresenta a interface do programa.

A seção (a) mostra as árvores de diretórios para *Seqüências (Sequence)*, *Padrões(patterns)* e *Amostras (samples)* contidas no arquivo da música (.s3m, .it, .xm, .mod). *Sequence* é um bloco numerado que contém um *pattern*. É usado para dar a forma³⁸ da composição. O *pattern* é o bloco que contém as notas sequenciadas, as informações de intensidades, duração, etc. é o .mid do arquivo. Os *samples* são as amostras de áudio que fazem o papel de instrumentos da música.

Em (b), vemos uma biblioteca de amostras que podem usadas para se criar os diferentes instrumentos do *Module File*, elas não fazem parte do arquivo final (.it, S3M, etc.). Em (c), temos as opções de pastas. A opção *General* diz respeito ao andamento e volume final.

³⁸ Forma é um termo técnico da Musical que se refere à organização das seções de uma música. Populamente se diz que uma música é formada por; introdução, verso 1, verso 2, refrão, etc.

A seção (d) é a aérea para a mixagem da música, onde serão acertados os volumes, posições estéreo, efeitos, etc.

No Quadro 3.2 destacamos a seção para edição de amostras do programa. Em (a) são acertadas as propriedades de loops, permitindo que a amostra seja lida do início até o final, em loop contínuo, com loop a partir de um ponto e com loop reverso. Em (b) visualiza-se a amostra e os resultados das transformações destes processos.

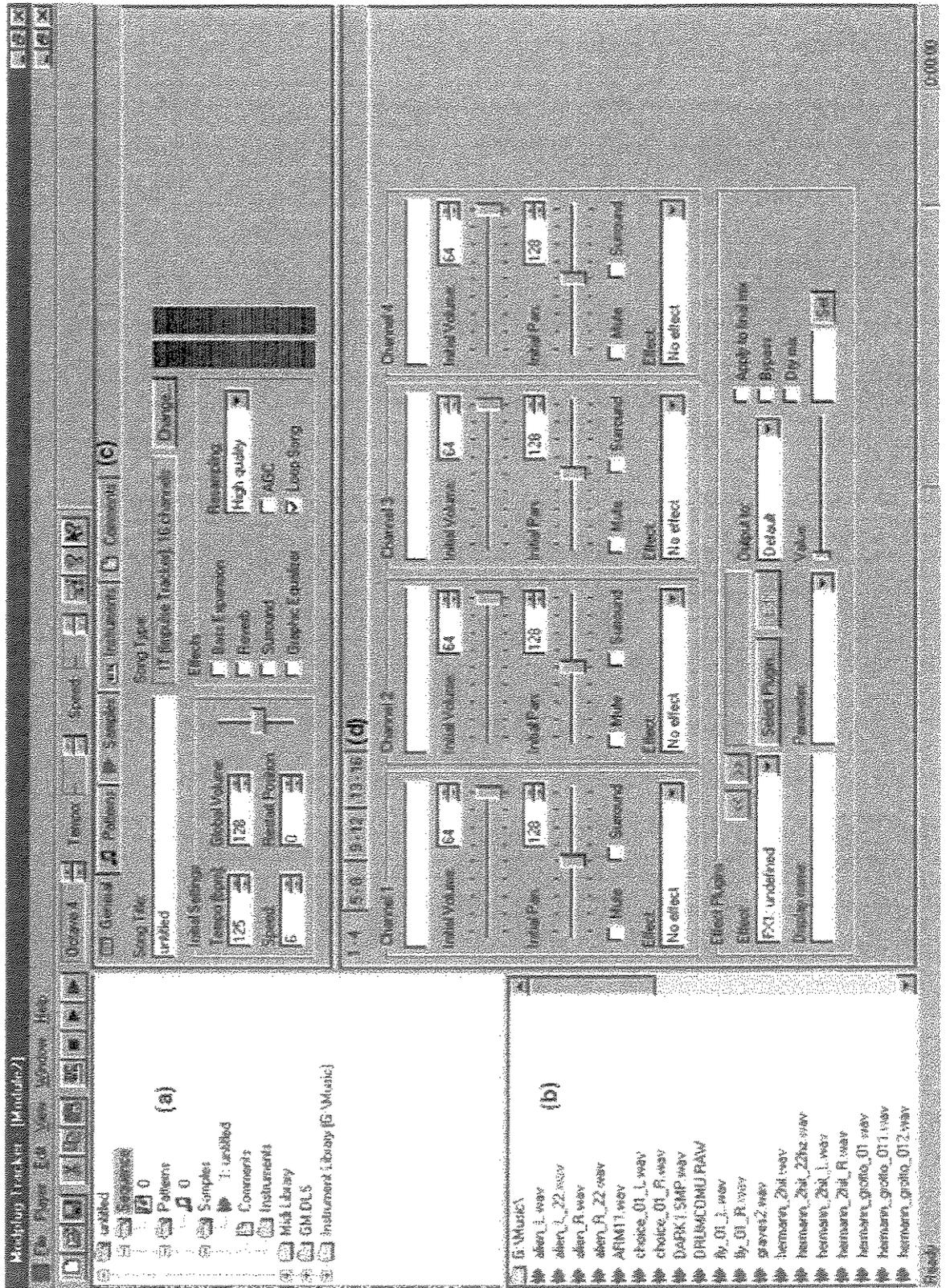
No Quadro 3.3 encontramos a seção para a edição da parte MIDI do programa. Em (a), temos o exemplo de um *pattern*. As informações de performance de cada instrumento são anotados em seus respectivos canais e seguem o molde a seção (b).

Analisado sob vários aspectos o Modplug Tracker traz várias vantagens para o compositor de músicas eletrônicas. Ele aceita que se trabalhe em mesmo arquivo com várias amostras de áudio, com taxas de amostragem e profundidade de bits variadas. Parece uma ferramenta poderosa, mas ela apresenta um problemas: a Interface.

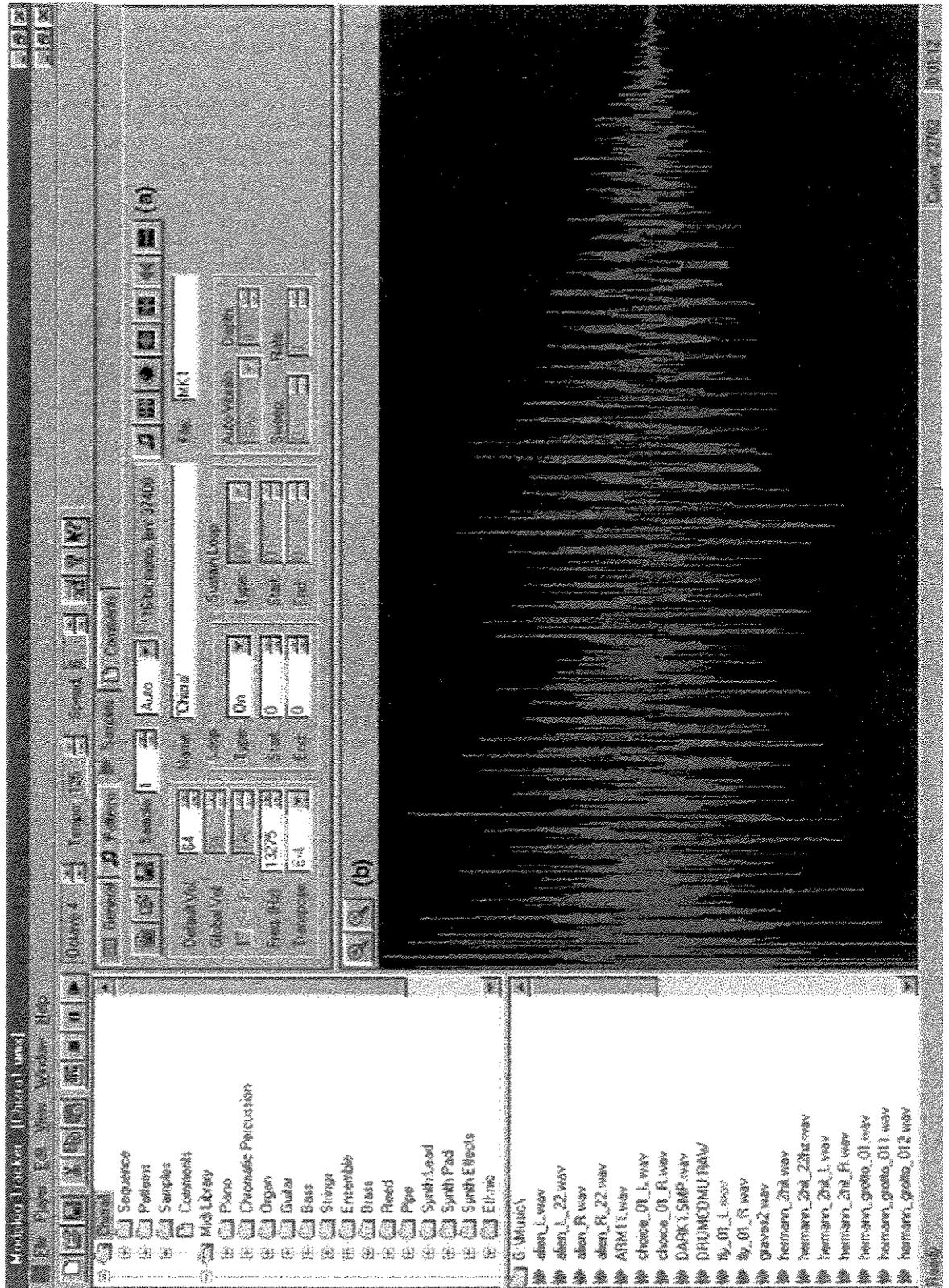
A interface de criação de música do programa está atrasada em relação aos produtos profissionais do mercado em pelo menos dez anos. Por exemplo, quando da gravação de um canal com instrumento MIDI, o programa só gravava as informações da nota musical, sua localização no compasso e duração, sem informações de dinâmica³⁹. Várias informações de performance têm de ser inseridas via teclado do computador, a duração correta das notas, o volume de cada uma, informações de vibratos, etc. É o equivalente a escrever um livro usando somente carimbos com as letras do alfabeto!

³⁹ Dinâmica: Intensidade sonora de uma nota musical. A variação de dinâmica é medida em dB (decibéis). No MIDI é o parâmetro *Velocity* que associa a velocidade de ataque da nota com sua intensidade.

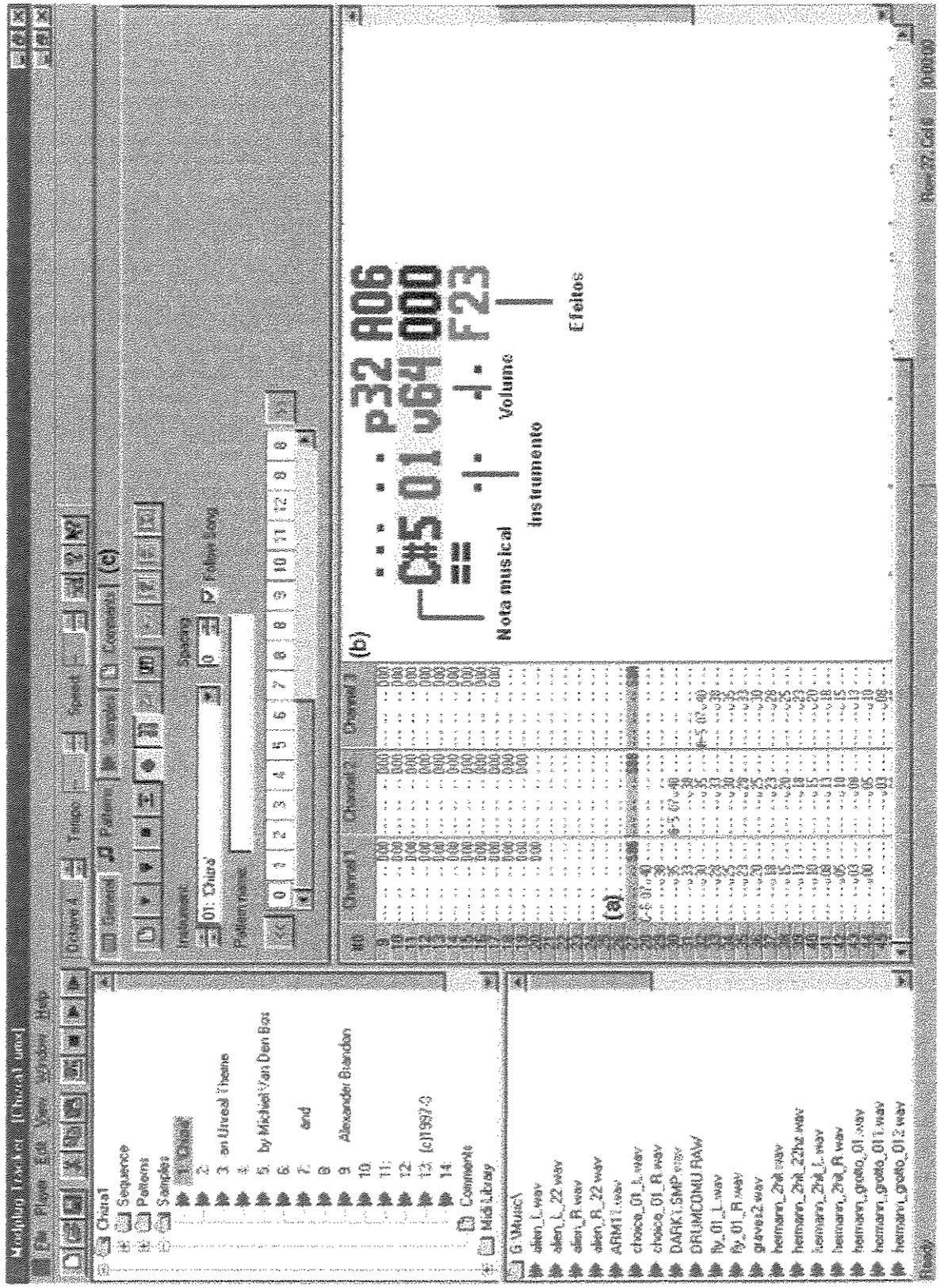
Quadro 3.1 - Interface do Modplug Tracker - Principal



Quadro 3.2 - Interface do Modplug Tracker - Amostras de Áudio



Quadro 3.3 - Interface do Modplug Tracker - Edição de Patterns



Situação de Contorno

Mesmo com todos estes defeitos era o único programa disponível, compatível com o UnrealEd. A trilha musical deveria ser feita normalmente nos padrões profissionais e de alguma forma deveria ser inserida no jogo.

A solução encontrada foi: transformar a música inteira em uma amostra “gigante” de áudio. E tratá-la no Modplug Tracker como um instrumento normal. Para tanto adotamos os seguintes procedimentos:

- O arquivo da música finalizada em qualidade de CD (44Khz, 16bits), era separado em dois arquivos: um contendo o canal esquerdo e o outro o canal direito.
- Estes dois canais mono eram reamostrados para 22 KHz e 16 bits, que era o formato com maior qualidade que o UnrealEd conseguia trabalhar.
- Em seguida, criava-se uma música em branco no Modplug com somente dois instrumentos (o canal esquerdo e o canal direito da música).

A música pronta fica somente com duas notas musicais que servem para disparar as amostras de áudio. Alguns ajustes na duração dos compassos também tinham de ser feitas para que a música fosse cortada durante sua execução.

Para obter um sinal com maior volume sonoro, foi necessário triplicar o número de canais usados. Portanto cada canal estéreo é formado pela soma de três canais mono.

Considerações Estéticas

O jogo Black Death, como a maioria dos jogos de ação, tem um forte caráter épico, o jogador em geral encarna um herói que terá uma jornada cheia de perigos e aventuras fantásticas. A grandiosidade dos atos deste herói são refletidas e insinuadas pela música. Os perigos são agigantados, os inimigos são terríveis e invencíveis.

A opção da trilha foi a de enfatizar os aspectos fantásticos/místicos que cercam a figura de um templário, criando ambiência, gerando envolvimento do jogador e promovendo sua imersão na trama do jogo.

Onde inserir a música

Saber onde inserir ou não a música faz parte da trabalho de qualquer compositor de trilhas. Uma trilha musical presente o tempo todo eqüivale a nenhuma trilha musical. A música perde seu sentido, não marca nenhum evento, não indica nenhum aspecto especial do ambiente, perde algumas de suas principais funções.

A partir da divisão da fase da catedral nas nove áreas: Cripta Templária, Catacumbas, Corredores, Capela subterrânea, Prova templária, Biblioteca, Subterrâneos, Nave Central e Torre. Foram selecionadas áreas que permitissem ao jogador explorar o ambiente por um certo tempo. Para que música fosse percebida como parte do “caminho” do jogador e não pertencente a um local estanque. Tais áreas são apresentadas na seqüência:

Início do jogo: Dentro da Cripta Templária

Um *trigger* localizado no início do jogo dispara o um tema rápido que sai dos metais e através de dobramentos chega até os contrabaixos. Este tema serve para dar o primeiro impacto do jogador, em sua saída da tumba. O trecho não dura mais de dez segundos e a partir daí o jogador só ouve os sons do ambiente. A região toda tem uma leve reverberação para diferenciar a arquitetura e o tipo do "material da construção" (mármore). Através desses elementos busca-se um certo grau de realismo sonoro no ambiente.

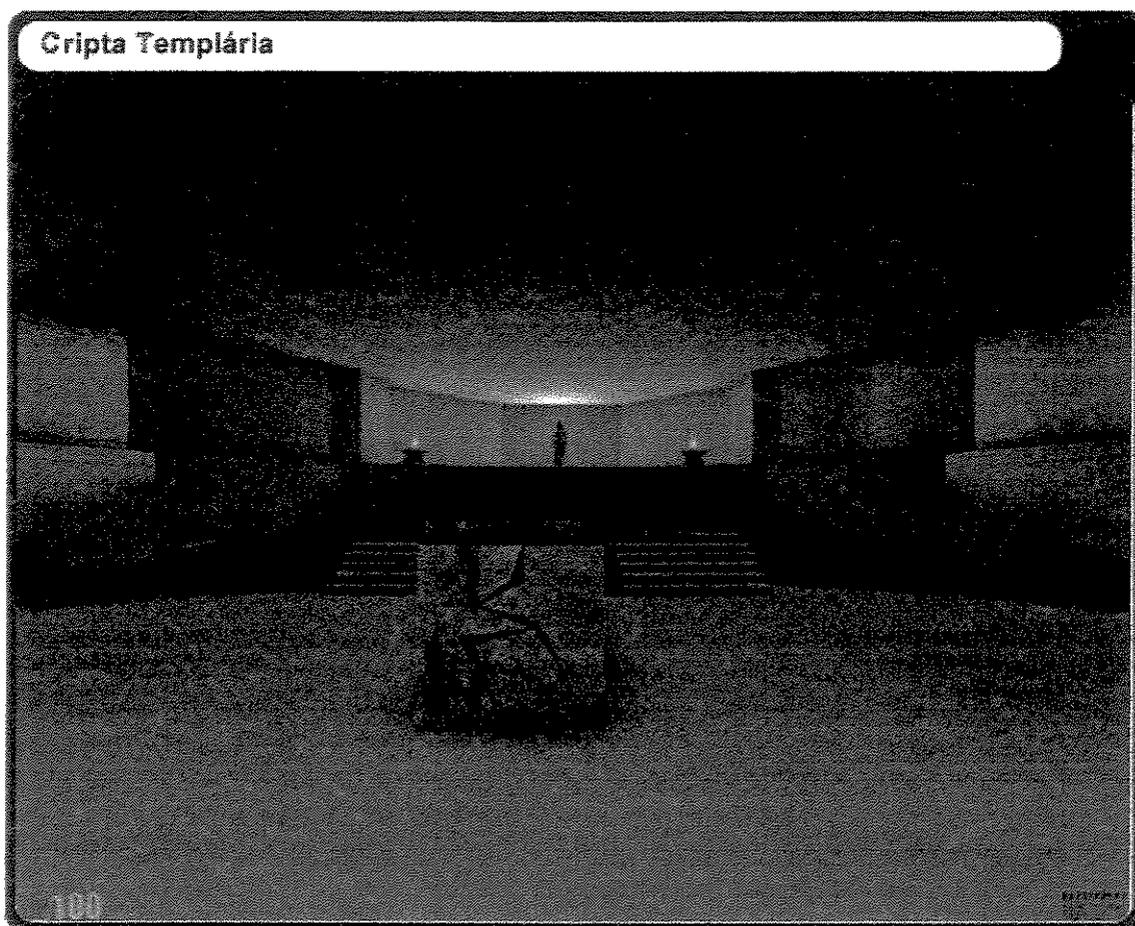


Figura 3.1 - Cripta Templária

Chegada às catacumbas

Após descobrir a saída cripta, o jogador tem de percorrer corredores e escadarias até encontrar uma passagem escondida para as catacumbas da catedral. Este o lugar mais sombrio de toda a fase e novamente a música acentua o primeiro olhar que o jogador lança sobre a seqüência de tumbas e cadáveres. Um *trigger* logo na entrada da catacumba dispara a música.

O tema tem um clima onírico e é conduzido por uma voz feminina em vocalise (sem o uso de palavras). Um ostinato⁴⁰ de Violoncelos e Fagotes e acompanham a música inteira. As acentuações da música são discretas para reforçar o aspecto contemplativo da cena. O trecho dura aproximadamente um minuto.

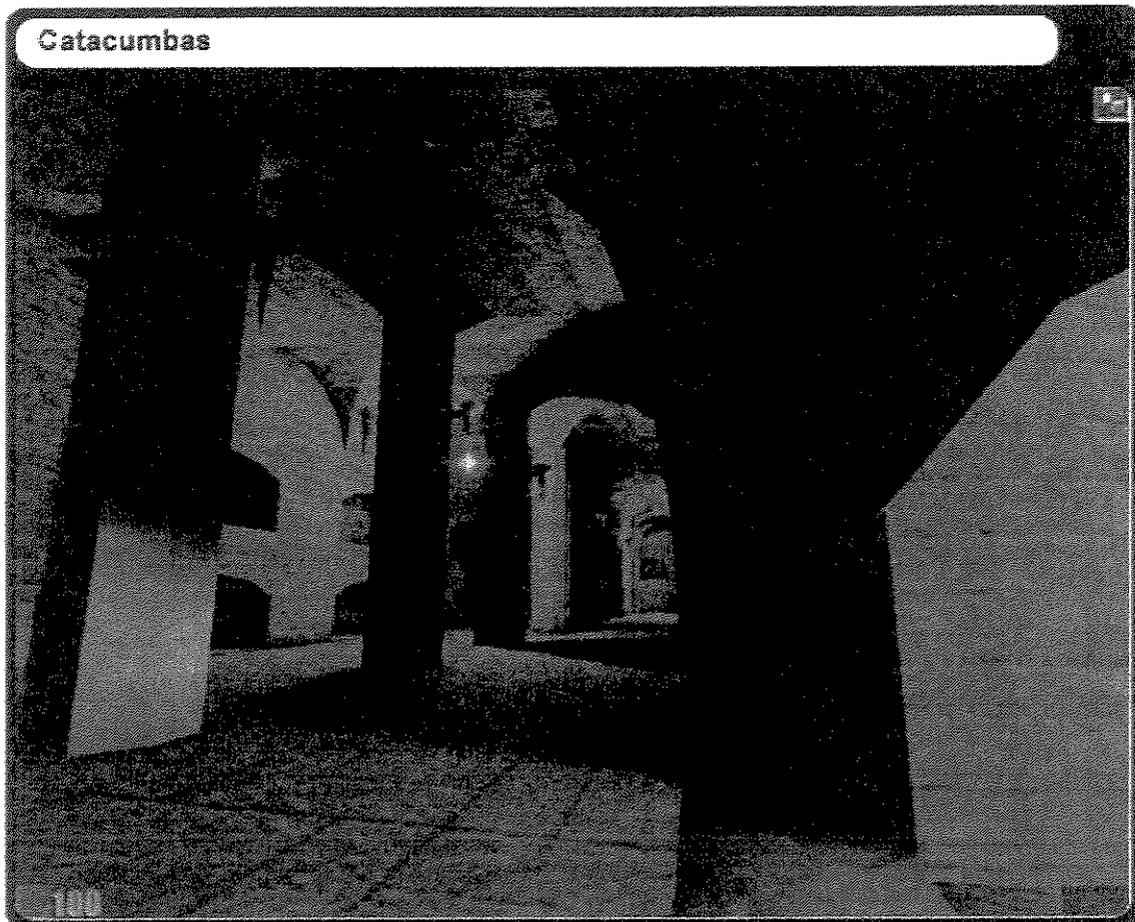


Figura 3.2 - Catacumbas

⁴⁰ O ostinato é figura ritmo-melódica que se repete indeterminadamente.

Corredores

Esta música inicia-se imediatamente após o final do tema das catacumbas. A passagem é imperceptível e é dependente somente do tempo, não depende da posição do jogador. Um *trigger* temporizado envia o evento que aciona a música.

Esta música funciona como uma transição entre a musica anterior e o som ambiente. É composta somente de um fundo muito grave com eventuais ruídos e silvos. Seu último compasso repete o ostinato do tema das catacumbas, servindo como ligação entre os dois momentos musicais.

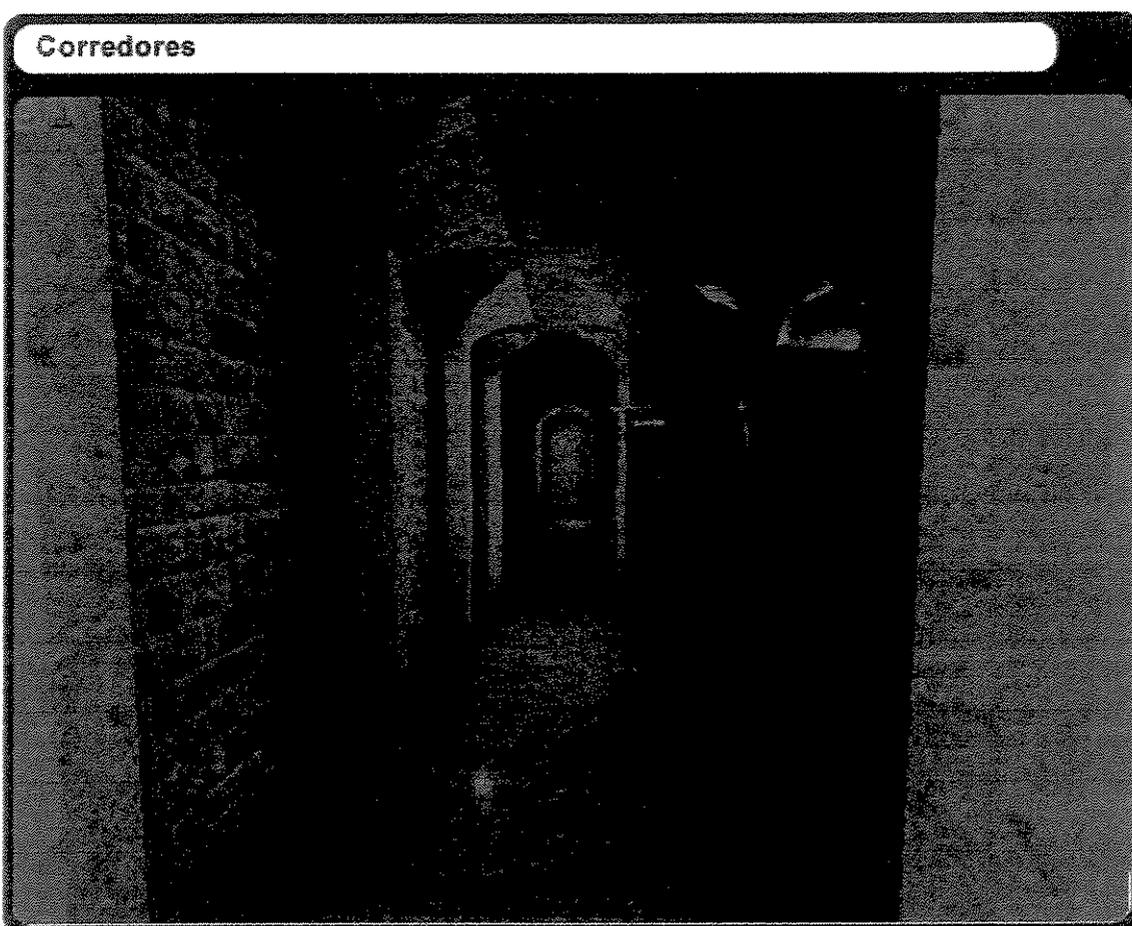


Figura 3.3 - Corredores

Prova Templária

O momento mais tenso do jogo é marcado por uma instrumentação “densa”, que utiliza todos os instrumentos da orquestra. Os contrabaixos e os tímpanos executam um *ostinato* em andamento rápido, com *preenchimentos* da caixa que executa células rítmicas militares. Todos os acentos são dados nos contratempos ora corda em *pizzicato* ora pelas trompas. É a música com tempos mais definidos do jogo e tentam-se realmente impor um ritmo para a ação do jogador. Na prova templária o jogador deve pular nas “pedras” certas enquanto o chão literalmente se desmancha. O *trigger* da primeira pedra dispara a música.

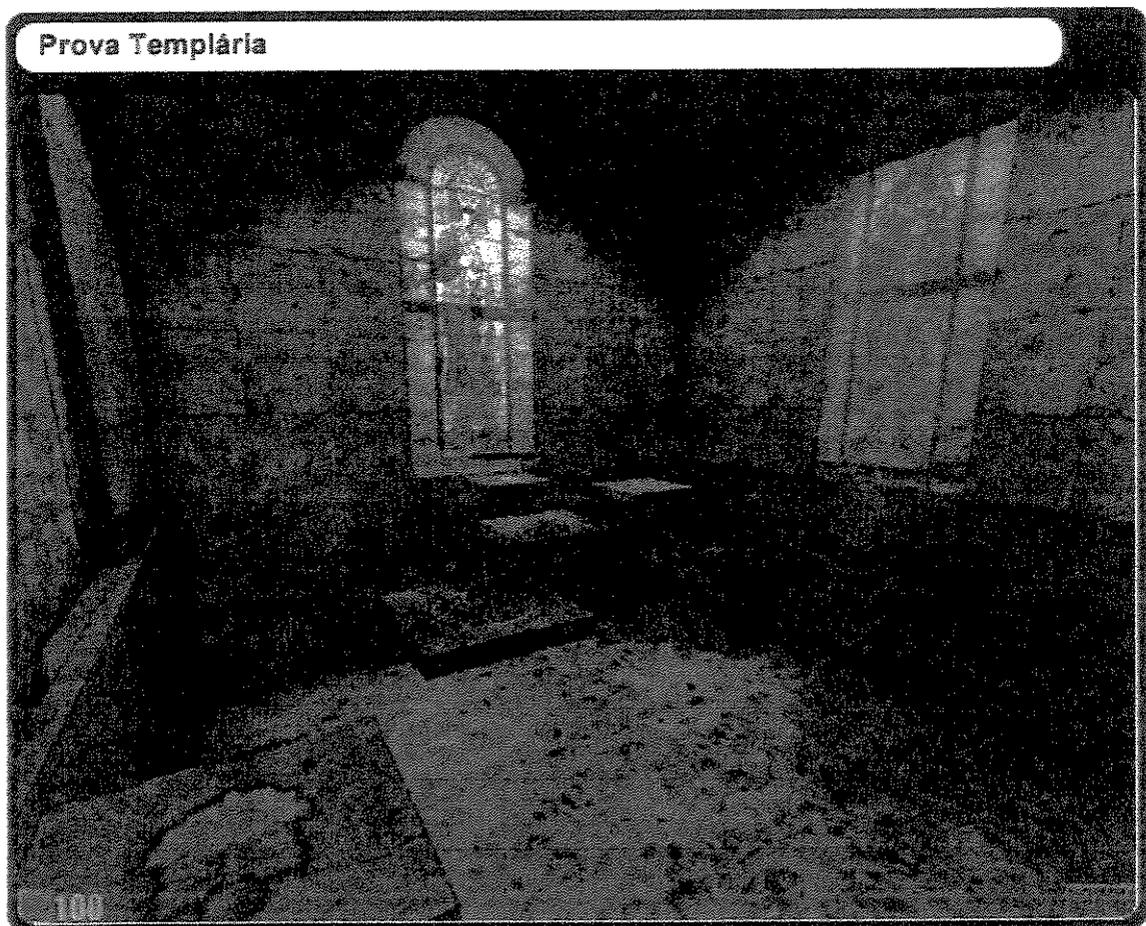


Figura 3.4 - Prova Templária

Torre

A torre é a última parte da catedral. O jogador chega ao fim das escadarias e percebe que ainda existe uma sala mais acima que não pode ser alcançada. A única saída é saltar. A música é contemplativa mais uma vez a voz feminina carrega o tema. A harpa reforça o caráter onírico da cena. Um *trigger* no último degrau da escada aciona a trilha.

O jogador salta. Se ele já havia encontrado a imagem do graal na biblioteca, recebe como prêmio o canto dos anjos e levita até a sala. Caso contrário despenca para a morte.

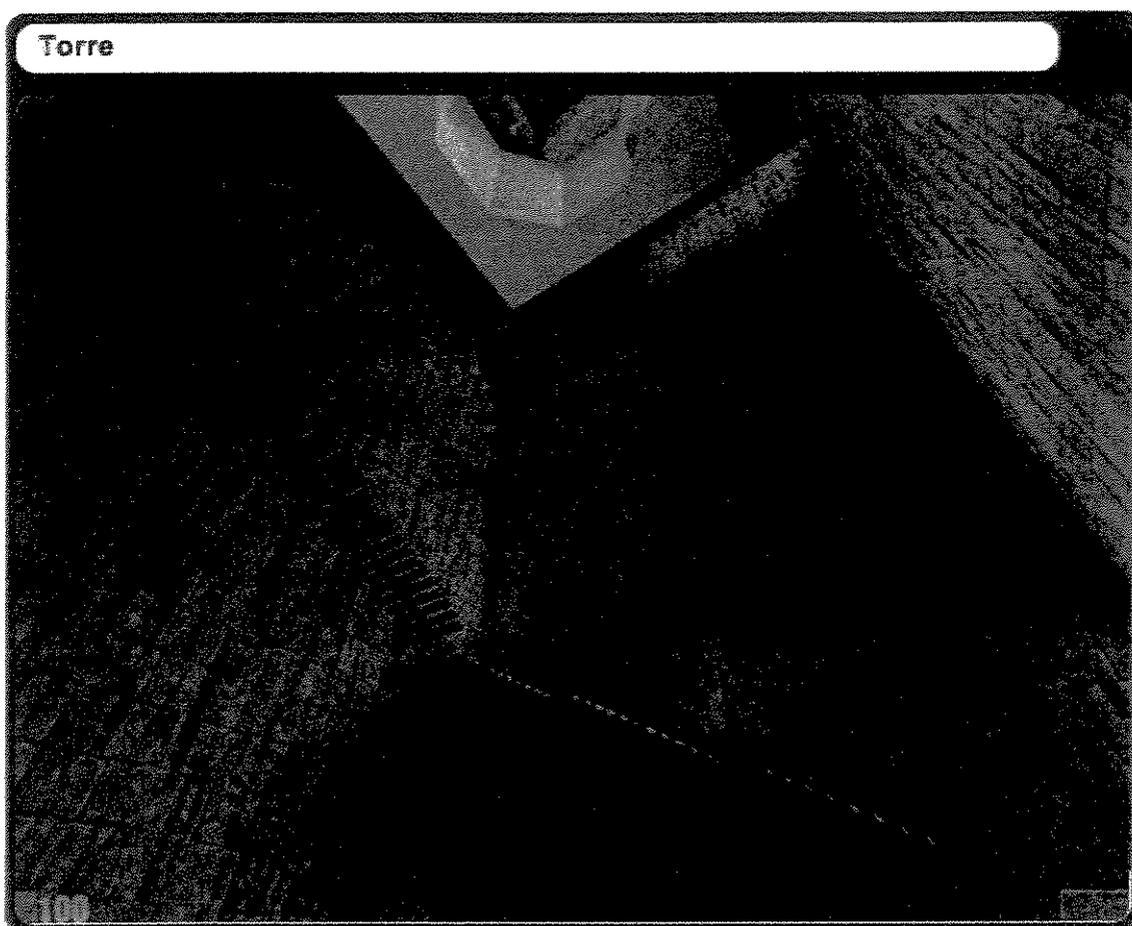


Figura 3.5 - Torre

Considerações Finais

Criar um jogo eletrônico com uma equipe de desenvolvimento de um homem só, não uma das tarefas mais usuais para um músico. Mas o gosto pelo desafio, pela busca de novos caminhos e parte integrante do perfil dos desenvolvedores de jogos.

Mais que um jogo finalizado Black Death foi aprendizado constante. Esperamos que ele tenha desmistificado o processo de criação de um jogo e que tenha mostrado que os “pré” conceitos não devem desviar o jogo de seu foco: o entretenimento.

Desejamos que experiência de se jogar Black Death, com este tipo de organização de trilha sonora seja mais rica e atraente do que as trilhas feitas nos moldes de muitos jogos de mercado.

Black Death (ou pelo menos parte dele) nem de longe encerra a questão. Poucos de seus resultados são mensuráveis. Mas sua intenção é mostrar que a mudança de conceitos é tão importante quanto desenvolvimento de novas tecnologia para a música nos jogos. O cinema já encontrou uma forma estruturada de articulação, já existem aqueles que falam em estagnação de linguagem, o jogo eletrônico é jovem, é uma campo para experimentações muito ricas para compositores, *sound designers* e profissionais do áudio.

As ferramentas ainda não estão desenvolvidas ou pelo menos maduras o suficiente para que possam romper o círculo dos especialista.

O futuro? Música Interativa

A palavra chave de Trilha Sonora Musical no cinema é: Sincronismo!!!! E no jogos? Provavelmente este conceito tem de ser revisado. A música interativa, e algumas experiências começam dar resultados. Várias empresas associadas a IASIG - Interactive Audio Special Interest Group, buscam soluções e ferramentas que permitam que a música possa ser feita conforme o jogador joga (on the fly), misturando tecnologias DLS(Downloadable Sample) e MIDI.

A própria Microsoft tem um programa gratuito que até agora só foi utilizado em experiência isoladas.

Logo a música nos jogos será mais interativa, mais provocativa e mais emocionante, assim como os próprios jogos.

Muitas conclusões e apontamentos sobre Design, Jogabilidade e Design Sonoro, de todo esse período de trabalho, ainda ficaram por serem ditos. Mas, logo encontrarão seu caminho na próxima jornada.

BIBLIOGRAFIA

MÚSICA

CARRASCO, Claudiney R. *Trilha musical: música e articulação fílmica*. Tese de Mestrado ECA/USP. São Paulo, 1993.

HOPPIN, Richard H. *Medieval Music*, London (WW – Norton and Company), 1978.

LANG, Paul Henry. *Music in Western Civilization*, London (WW – Norton and Company), 1997.

MICHELS, Ubrich. *Atlas da Música I*, Madrid (Alianza Editorial), 1982.

PRENDERGAST, Roy M. *Film Music – A Neglected Art*, New York, (WW – Norton and Company), 1977.

ARQUITETURA

ADAM, Ernest. *Arquitetura Medieval*, Lisboa (Verdo), 1970.

DESPRA, Pierre. *Notre-Dame de Paris*, Paris (Ed. Barthaud), 1939.

JORDAN, R. Forneaux. *Western Architecture*, London (Thames and Hudson Ltd.), 1996.

SIMSON, Otto George Von. *The Gothic Cathedral: Origins of Gothic Architecture and the Medieval Concept of Order*, Princeton (Princeton University), 3rd ed., 1988.

MACAULAY, David. *Construção de uma Catedral*, Martins Fontes 1^a ed., 1988.

REICH, Ludwig Heiyden e Lotz, Wolfgang. *Arquitetura en la Italia 1400-1600*, Madri (Edições Catedra), 1974.

RORIMER, James J. *The Cloisters – The Building and the Collection of Medieval Art*, New York (The Metropolitan Museum of Art), 1939.

BIOLOGIA

BURNETT, George W.; Scherp, Henry; Schuster, George S. *Microbiologia Oral e Doenças Infeciosas*, Rio de Janeiro (Ed. Guanabara Koogan), 4ª ed., 1978.

HUBBERT, Willian T.; McCulloch, Willian F.; Schnurrenberger, Paul R. *Diseases Transmitted from Animals to Man*, Spring Field (Charles & Thomas Publishers), 6th ed., 1975.

GERAL E COMUNICAÇÃO

CASTLENOVI, Gia Vittorio; Negri, Carlo de; Elia, Olga; Mazzino, Edoardo; Pistarino, Geo; Raimond, Piero, Scotti, Pietro. *Liguria*, Roma (Banca Nazionale del Lavoro), 1967.

CHATMAN, Seymor. *Story and Discourse/Narrative Structure in Fiction and Film*, Ithaca (Cornell University Press), 1978.

DORFLES, Gillo. *Novos ritos, novos mitos*. Lisboa: Edições 70, 1965.

FOLDA, Jarolav. *The Art of the Crusades in the Holy Land 1098-1197*, Chapel Hill (Cambridge Press), 1995.

MARTINDALE, Andrew. *Gothic Art*, London (Thames and Hudson Ltd.), 1994.

LÉVY, Pierre. *As tecnologias da inteligência*. São Paulo: Editora 34, 1995.

MCLUHAN, Marshall. *Os meios de comunicação como extensões do homem*. São Paulo: Editora Cultrix, 1996.

PROPP, Vladimir Iakovlevich. *As Raízes Históricas do Conto Maravilhoso*, São Paulo (Ed. Martins Fontes), 1997.

PROPP, Vladimir Iakovlevich. *Morfologia do Conto Maravilhoso*, Rio de Janeiro (Ed. Forense - Universitária), 1984.

ROSENFELD, Anatol. *O teatro épico*. São Paulo: Editora Perspectiva, 1985.

DESIGN DE JOGOS E ENGENHARIA DE SOFTWARE

ADAMS, Ernest . Dogma 2001: A Challenge to Game Designers. Gamasutra. 2, de fevereiro de 2001.

http://www.gamasutra.com/features/20010202/adams_01.htm

BRANDON, Alexander. Interactive Music: Merging Quality with Effectiveness. Gamasutra. 27 de março de 1998.

http://www.gamasutra.com/features/sound_and_music/19980327/interactive_music_quality_intro.htm

BROWN, Duncan. GDC 2001: Current Architecture and Potential Approaches to Level Design. Gamasutra. 24 de março de 2001.

http://www.gamasutra.com/features/20010324/brown_01.htm

CHEN, Steven and Brown, Duncan. The Architecture of Level Design. Gamasutra. 16 de julho, 2001.

http://www.gamasutra.com/features/20010716/chen_01.htm

CRAWFORD, Chris. *The Art of Computer Game Design*, Vancouver (Washington State University Vancouver), 1982.

HARLAND, Kurt. Composing for Interactive Music. Gamasutra. 17 de fevereiro de 2000.

http://www.gamasutra.com/features/20000217/harland_01.htm

HELANDER, M.G.; LANDAEUR T.K.; PRABHU, P.V. Handbook of human-computer interaction. Elsevier: North Holland, 2 ed, 1997.

KREIMEIER, Bernd. The Case For Game Design Patterns. Gamasutra
13 de março, 2002.

http://www.gamasutra.com/features/20020313/kreimeier_01.htm

LITTLEJOHN, Randy. The Need to Adapt the Tools of Drama to Interactive Storytelling.
Gamasutra. 14 de setembro, 2001.

http://www.gamasutra.com/features/20010914/littlejohn_01.htm

MALONE, Thomas Wendel. *What Makes Things Fun to Learn? A Study of Intrinsically Motivating Computer Games*, (Stanford University), 1980.

MELAX, Stan. GDC 2001: BSP Collision Detection As Used In MDK2 and NeverWinter Nights. Gamasutra. 24 de março de 2001

http://www.gamasutra.com/features/20010324/melax_01.htm

NORMAN, Donald A. *The Invisible Computer - Why good products can fail, the personal computer is so complex, and Information Appliances are the solution*. Cambridge, Massachusetts; London, England. The MIT Press, 1998.

OLIVEIRA, L. O; BARANAUSKAS, M.C.C. Semiotic proposals for software desing: Problems and prospects. Relatório técnico. IC-98-10. Abril de 1998. Instituto de Computação - Unicamp.

WINOGRAD, T. in *Bringing Design to Software*. New York: Addison-Wesley, 1996.

PAGÁN, Tito. Where's the Design in Level Design? Part One. Gamasutra. 16 de julho, 2001.

http://www.gamasutra.com/features/20010716/pagan_04.htm

PAGÁN, Tito. Where's the Design in Level Design? Part Two. Gamasutra. 16 de julho, 2001. http://www.gamasutra.com/features/20010716/pagan_04.htm

ROCHA, Heloísa Vieira e Baranauskas, Maria Cecília Calani. *Design e Avaliação de Interfaces Humano-Computador*, São Paulo (IME-USP), 2000.

ROSS, Rob. Interactive Music...er, Audio. Gamasutra. 15 de maio de 2001

http://www.gamasutra.com/resource_guide/20010515/ross_01.htm

ROUSE III, Richard. Game Design: Theory and Practice, Chapter Seven - The Elements of Gameplay. Gamasutra. 27 de junho, 2001

http://www.gamasutra.com/features/20010627/rouse_01.htm

SHELLEY, Bruce. Guidelines for Developing Successful Games. Gamasutra. 15 de agosto de 2001.

http://www.gamasutra.com/features/20010815/shelley_01.htm

ZIZZA, Keith. Your Audio Design Document: Important Items to Consider in Audio Design, Production, and Support. Gamasutra. 26 de julho de 2000.

http://www.gamasutra.com/features/20000726/zizza_01.htm

História dos jogos Eletrônicos

www.gamespot.com

www.outerspace.terra.com.br

<http://www.emuunlim.com/doteaters/>

Documentação de Suporte (jogos)

Aliens Versus Predator. Rebellion/Fox Interactive, 1999.

Gabriel Knight II - The beast within. Sierra, 1994.

Half Life. Valve, 1998.

MDK- Muder, Death and Kill. Shine, 1996.

Mech Warrior II. Activion, 1995.

Quake, ID Software, 1996.

Quake II, ID Software, 1997.

Quake III – Arena, ID Software, 2000.

Rebel Assault II - The hidden empire. LucasArts, 1994.

Redneck Rampage. Interplay, 1997.

Star Wars Episode I – The Phantom Menace. Lucas Arts, 1999.

The Need For Speed SE. Eletronic Arts, 1996.

Unreal. Epic Games, 1998.

Unreal Tournament. Epic Games, 1999.

Vampire the Masquerade –Redemption. Nihilistic, 2000.

X-wing vs. Tie Fighter. LucasArts, 1997.

Wing Commander III. Origin, 1994.