

André Luiz Gomes dos Santos Filho, Daniel Carvalho Climaco, Davi André De
Carvalho Pereira, Miguel Büge Paganini, Miguel Saiter Padovan

Análise Técnica de Wolfenstein 3D

Vila Velha - ES

2026

André Luiz Gomes dos Santos Filho, Daniel Carvalho Climaco, Davi André De
Carvalho Pereira, Miguel Büge Paganini, Miguel Saiter Padovan

Análise Técnica de Wolfenstein 3D

Trabalho acadêmico apresentado como requi-
sito para avaliação.

Universidade de Vila Velha
Curso de Ciência da Computação

Vila Velha - ES
2026

Sumário

0.1	Mapas	3
0.2	Áudio	4
0.2.1	Sons	4
0.2.2	Música	5
0.3	Distribuição	7
0.4	Raycasting	8

0.1 Mapas

Os mapas foram desenvolvidos com o auxílio de um editor interno denominado TED5 (Tile Editor). Embora não tenha sido originalmente projetado para Wolfenstein 3D, o TED5 surgiu durante o desenvolvimento da série Commander Keen e foi sendo progressivamente aprimorado ao longo do tempo.

Tratava-se de uma ferramenta altamente flexível, capaz de suportar tanto jogos do tipo side-scroller quanto títulos com perspectiva superior.

O TED5 não operava como uma aplicação independente. Sua execução dependia da presença de um arquivo de recursos e de seu respectivo cabeçalho, conforme descrito no fluxo de manipulação de recursos gráficos.

Nesse contexto, os identificadores de texturas eram diretamente incorporados à estrutura dos mapas, evidenciando uma forte integração entre o editor e o pipeline de dados do jogo.



Figura 1 – Exemplo de mapa



Figura 2 – Exemplo de mapa

O editor possibilitava a organização dos elementos do mapa em camadas denominadas “planos”. Essa abordagem em camadas demonstrou-se extremamente poderosa.

Em Commander Keen, por exemplo, diferentes planos eram utilizados para representar o fundo, plataformas transitáveis, itens coletáveis, entre outros.

Em Wolfenstein 3D, a estrutura foi simplificada para dois planos principais: um dedicado às paredes e outro responsável por entidades interativas, como inimigos e bônus.

A reutilização do TED5 trouxe benefícios significativos. Além da economia de tempo no desenvolvimento de ferramentas, houve uma redução considerável na curva de aprendizado, uma vez que a equipe já dominava o editor há anos. Essa familiaridade permitia a criação de níveis em um intervalo extremamente curto, frequentemente em questão de minutos.

Estimativas indicavam que um nível completo poderia ser produzido em aproximadamente um dia, permitindo a expansão do jogo em múltiplos episódios.

0.2 Áudio

0.2.1 Sons

No início da década de 1990, o ecossistema de hardware de áudio para PCs era altamente fragmentado. Diante desse cenário, foi adotado suporte a múltiplos dispositivos, incluindo diferentes placas de som e o alto-falante interno do computador. Essa decisão implicava a necessidade de gerar múltiplas versões dos mesmos efeitos sonoros.

Para gerenciar os dados de áudio, utilizou-se a ferramenta MUSE, responsável por agrupar e compactar os recursos em um formato proprietário armazenado no arquivo AUDIOT.



Figura 3 – MUSE splash screen

Cada efeito sonoro era disponibilizado em três variações:

- Compatível com o alto-falante do PC
- Adaptado para placas AdLib
- Otimizado para Sound Blaster

Essa abordagem evidencia a preocupação com portabilidade e compatibilidade em nível de hardware.

0.2.2 Música

As composições foram produzidas por Bobby Prince utilizando ferramentas como Sequencer Plus Gold e Cakewalk.

O processo envolvia:

1. Criação inicial no Cakewalk
2. Exportação para formato MIDI
3. Importação no SPG para ajuste de instrumentos OPL

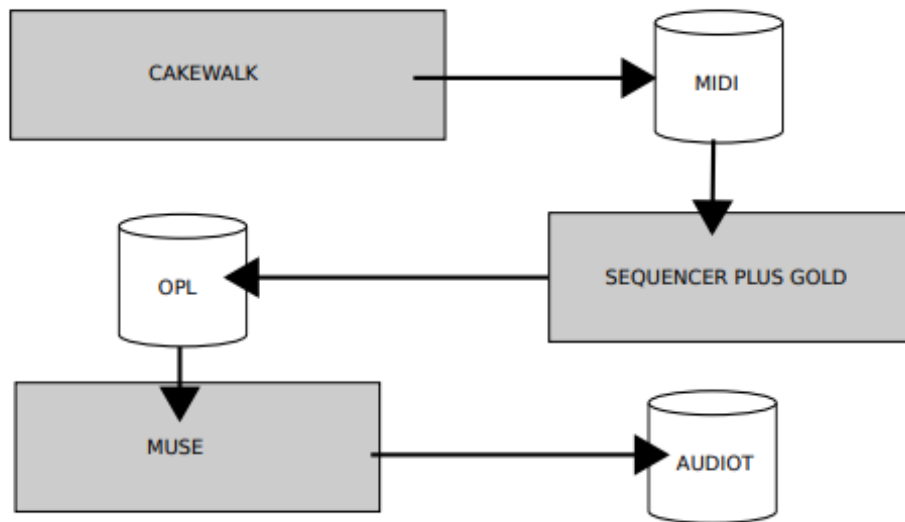


Figura 4 – Fluxo de trabalho musical descrito por Bobby Prince

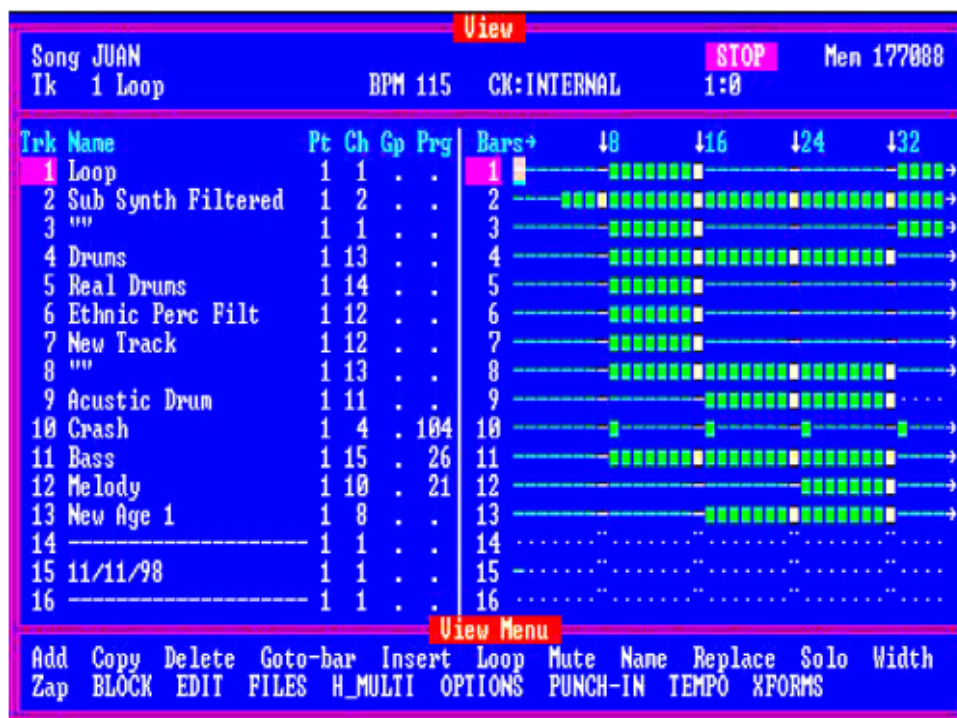


Figura 5 – Sequencer Plus Gold (“SPG”) da Voyetra

Os arquivos eram convertidos para o formato IMF, projetado para o chip YM3812 (OPL2). Esse formato não utilizava abstrações de alto nível; em vez disso, consistia em uma sequência de instruções em nível próximo ao hardware

O chip operava com nove canais independentes, cada um utilizando dois osciladores (modulador e portador).

Essa arquitetura permitia uma ampla variedade de timbres, com forte identidade sonora baseada em funções senoidais.

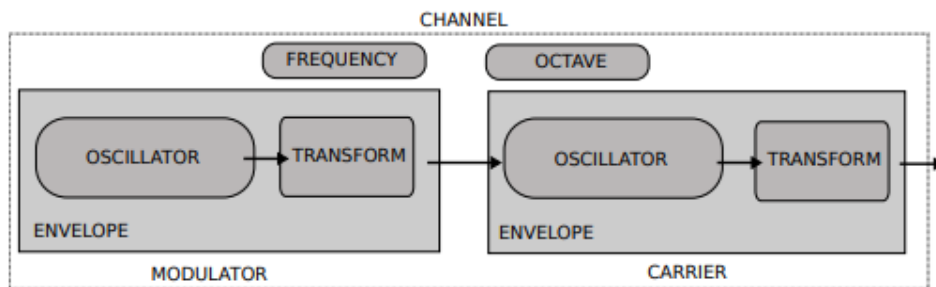


Figura 6 – Arquitetura de um canal YM3812

0.3 Distribuição

O lançamento inicial ocorreu em 5 de maio de 1992 via BBS, utilizando o modelo shareware, no qual o primeiro episódio era disponibilizado gratuitamente, enquanto os episódios adicionais exigiam pagamento.

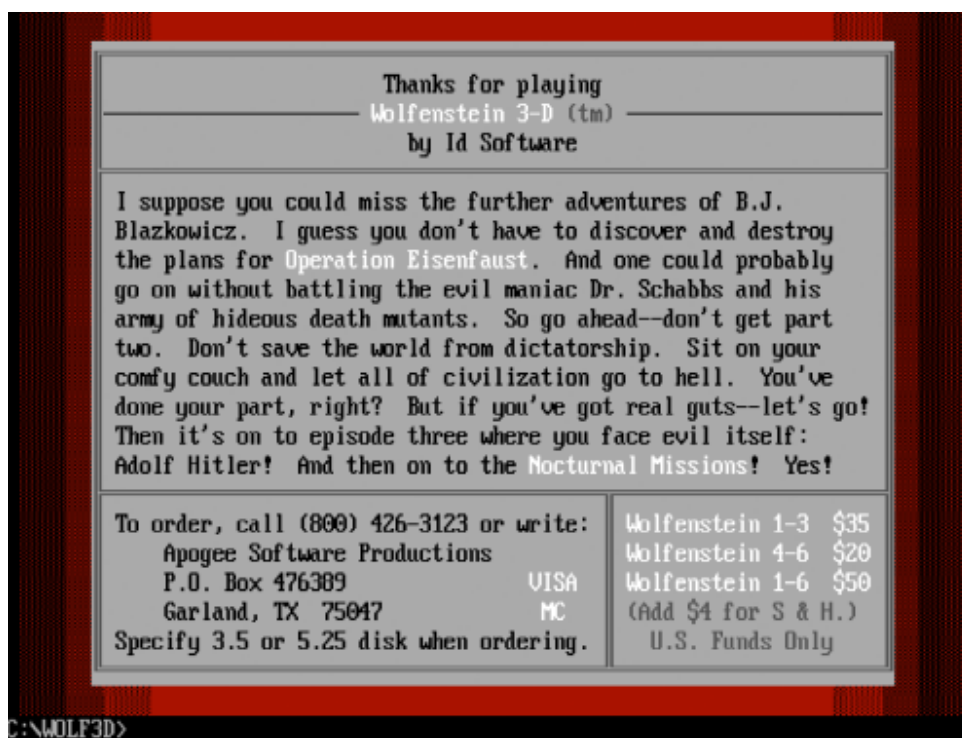


Figura 7 – A saída do jogo descreve como obter a versão completa.

Devido às limitações da época, o principal meio de distribuição era o disquete de 3½ polegadas.

Os dados foram comprimidos de mais de 1 MB para aproximadamente 645 KiB, permitindo distribuição eficiente.

A estrutura de arquivos incluía:

- Executável principal

- Arquivos de recursos
- Dados de áudio
- Estruturas de mapas
- Interface gráfica comprimida

O executável utilizava compressão LZEXE, reduzindo significativamente seu tamanho.

O meio físico de distribuição — o disquete — possuía características relevantes do ponto de vista de hardware, como mecanismos de proteção contra escrita e diferenciação entre capacidades de armazenamento, aspectos diretamente relacionados à arquitetura de sistemas de armazenamento da época.

Em 1995, o jogo foi relançado em CD-ROM, acompanhando a evolução tecnológica.

0.4 Raycasting

O núcleo do motor gráfico de Wolfenstein 3D baseia-se na técnica de raycasting, eficiente para simular ambientes tridimensionais em hardware limitado, sem a necessidade de uma pipeline gráfica 3D completa.

Diferente da renderização 3D moderna, o raycasting opera em um espaço bidimensional e projeta a visão em primeira pessoa através do lançamento de raios.

Para cada coluna da tela:

- Um raio é emitido
- Detecta-se a interseção com uma parede
- A distância define a altura renderizada

Esse modelo permite transformar um problema de renderização 3D em uma série de cálculos 2D relativamente simples, reduzindo drasticamente o custo computacional.

As principais vantagens incluem:

- Baixo uso de memória
- Acesso previsível à memória
- Processamento sequencial
- Ausência de cálculos 3D complexos

Além disso, o motor utilizava otimizações como:

- Aritmética de ponto fixo em vez de ponto flutuante

- Tabelas pré-calculadas para funções trigonométricas
- Redução de divisões, substituindo-as por multiplicações sempre que possível

Essa abordagem foi fundamental para permitir que o jogo rodasse em CPUs da época, como o Intel 80386, sem aceleração gráfica dedicada.