



**UNIVERSIDADE DO SUL DE SANTA CATARINA**  
**GABRIEL PAES, TIAGO BOEING, TIAGO CUNHA**

**IA EM HALF-LIFE**  
**MODELAGEM DE JOGOS DIGITAIS**

**TUBARÃO**

**2018**

## RESUMO

Praticamente todos os jogos utilizam em seu desenvolvimento alguma forma de Inteligência Artificial. A aplicação destas técnicas não se limita apenas ao cenário de games atuais, mas amplamente desde o início do desenvolvimento dos mesmos.

Na maior parte das vezes a IA está presente no controle do comportamento de personagens não controlados pelo jogador, conhecidos por (nonplayer characters - NPCs), seriam os obstáculos/inimigos durante a partida.

Palavras-chave: **Half-Life, Valve, Jogos, Inteligência Artificial.**

## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>5</b>
DINÂMICA DO JOGO	5
O QUE É IA?	6
POR QUE UTILIZAR IA?	6
<b>IA NO JOGO</b>	<b>7</b>
TÉCNICAS UTILIZADAS	7
FUNCIONAMENTO DE UMA FSM	8
ESTRUTURA DE CÓDIGO	9
MÁQUINA DE ESTADOS EM HALF-LIFE	9
PISCAR OLHOS E LOCALIZAR PLAYER	10
DISTÂNCIA DO ALVO E DISPARO DE SONS (GRITOS)	10
<b>CONCLUSÃO</b>	<b>12</b>
<b>REFERÊNCIAS</b>	<b>13</b>
<b>LISTA DE ILUSTRAÇÕES</b>	<b>14</b>

# 1 INTRODUÇÃO

O primeiro jogo desenvolvido pela Valve foi Half-Life, empresa fundada em 1996. Uma mistura de terror e ação baseado na Engine de Quake. Porém durante o desenvolvimento do jogo cerca de setenta por cento da engine teve de ser reescrita. Por ser um estúdio novo, muitas dificuldades foram encontradas, inclusive para a parceira juntamente a uma distribuidora (Sierra), pois se tratava de um projeto ambicioso para a época. Felizmente a distribuidora tinha planos de lançar um jogo de ação 3D e acabou assinando contrato de distribuição junto a Valve.

Half-Life foi apresentado pela primeira vez ao público em 1997 na E3<sup>1</sup> e encantou a todos, o jogo tinha lançamento previsto para o fim de 1997, porém a data de lançamento foi adiada várias vezes durante 1998, antes do jogo ser finalmente lançado em novembro do mesmo ano. Na E3 1998 ele ganhou o Game Critics Awards de Melhor Jogo de PC e Melhor Jogo de Ação.

O nome original do projeto do jogo era Quiver, inspirado na base militar Arrowhead, do livro de Stephen King, The Mist, livro esse que serviu de inspiração para o enredo do jogo. Gabe Newell explicou que o nome de Half-Life foi escolhido porque era sugestivo ao tema, não era clichê, e tinha um símbolo visual correspondente: a letra grega  $\lambda$  (lambda minúscula), que representa meia-vida na física nuclear. Segundo um dos designers do jogo, Harry Teasley, Doom foi uma grande influência para a maioria da equipe que trabalhou em Half-Life. Eles queriam Half-Life assustasse os jogadores, assim como Doom o fez. < <http://www.neogamer.com.br/2011/06/unseen-o-desenvolvimento-de-half-life.html> >

Foi também Half-Life que originou o popular jogo Counter Strike.

## 1.1 DINÂMICA DO JOGO

Encarne Gordon Freeman. Você começa o jogo em um trem, com destino a Black Mesa, um complexo localizado no Novo México, EUA, onde várias pesquisas científicas de todos os gêneros são realizadas. Você irá encarnar Gordon Freeman, um cientista formado pelo MIT (Instituto de Tecnologia de Massachusetts), que trabalha na área de pesquisa de materiais anômalos e segue para um de seus trabalhos mais importantes de sua carreira, uma experiência com um cristal misterioso que fora levado a Black Mesa.

Com alguns pequenos empecilhos no caminho de Gordon, como alguns cientistas duvidando de sua capacidade, o herói finalmente chega ao local da experiência. Quando tudo parece estar indo bem, algo dá errado no experimento com o cristal e diversas explosões começam a ocorrer em Black Mesa, o que acaba abrindo vários portais por todo o complexo, possibilitando a invasão de seres alienígenas de outra dimensão, dando início a uma das aventuras mais eletrizantes do mundo dos games.

Tudo isso se passa no controle e perspectiva do herói sem nunca tirar o controle do jogador. Em uma época em que os jogos de tiro em primeira pessoa apenas limitavam-se a copiar as fórmulas de Quake e Doom, Half-Life inovou e trouxe uma nova maneira de trabalhar no gênero. Chega de cenas de corte, histórias banais, ambientações demoníacas ou simples tiroteios sem muitas explicações, tudo isso é deixado para trás para entrar em cena uma nova maneira de encarar os fps, com uma ambientação realista, história empolgante e

---

<sup>1</sup> Electronic Entertainment Expo, mais conhecida como E3, é uma feira internacional dedicada a jogos eletrônicos. É considerada a mais importante do gênero, por reunir novidades relativas a lançamentos e tendências de mercado de várias empresas do setor dos jogos.

trazendo mais perguntas que respostas, puzzles, momentos de muita ação com tiroteios e fugas frenéticas, tudo isso mesclando com elementos de exploração, aventura e até mesmo terror, com os marcantes alienígenas proporcionando um ar sombrio em dados momentos da aventura de Freeman. < <http://gamehall.uol.com.br/v10/half-life/> >

## 1.2 O QUE É IA?

Inteligência Artificial (IA) é um ramo da ciência da computação que se propõe a elaborar dispositivos que simulem a capacidade humana de raciocinar, perceber, tomar decisões e resolver problemas, enfim, a capacidade de ser inteligente. < <https://www.tecmundo.com.br/intel/1039-o-que-e-inteligencia-artificial-.htm> >

### 1.2.1 POR QUE UTILIZAR IA?

Usar inteligência artificial é a diversão. Sua importância é quanto aos resultados que o sistema irá gerar, e não como o sistema chega até os resultados; ou seja, o problema não é como o sistema pensa, mas sim como ele age. Isso se deve pelo fato que jogos eletrônicos são negócios – os consumidores desses produtos os compram em busca de diversão, e não lhes interessa como a inteligência de um personagem no jogo foi criada, desde que ela transforme o jogo divertido e desafiador, além, claro, de tomar decisões coerentes com o contexto do jogo (TOZOUR, 2002) (SCHWAB, 2004).

## 2 IA NO JOGO

Um detalhe curioso acerca da IA do jogo envolve a intuição de Gabe Newell um dos idealizadores do projeto. Ao se aproximar do fim do projeto, Newell sentia que faltava algo, então mobilizou seu estúdio para criar uma fase que envolvesse tudo o que já estava pronto no projeto, incluindo inimigos, armas, eventos de script e level design, após esse teste toda a IA e cenários do jogo foram refeitos.

Utilizando uma IA baseada em scripts, Half-Life foi considerada a melhor aplicação de IA nos jogos eletrônicos da época, segundo Kishimoto (2004). Para Tozour (2002), esses scripts determinam a forma que um agente inteligente deve agir diante de diferentes situações. Quando foi lançado, o jogo recebeu aclamação da crítica - possibilitado pela inteligência artificial subjacente. Esse tipo de impacto na comunidade de IA do jogo está entre os mais influentes de todos os tempos. < <http://aigamedev.com/open/article/half-life-sdk/> >

### 2.1 TÉCNICAS UTILIZADAS

Em jogos de tiro em primeira pessoa (FPS) como é o caso de Half-Life a IA ficou conhecida pelo excelente nível tático dos inimigos, desenvolvida através do uso de máquinas de estado finito (Finite State Machines ou FSM) e scripts que determinam como um agente inteligente deve agir em várias situações (WOODCOCK 1999) (TOZOUR, 2002).

Mesmo após dez anos, olhar para a base de código do game é uma ótima maneira de aprender sobre como projetar sistemas de IA simples, mas eficazes. Toda a lógica da IA é codificada em C++ e não é excessivamente orientada a objetos, por isso é mais fácil de acompanhar do que mecanismos recentes baseados em dados (embora não seja tão simples de estender).

Para representar os objetivos dos inimigos no jogo, foram utilizadas máquinas de estado. Que nada mais são do que os estados em que os personagens podem se encontrar durante o jogo, o atual estado onde o usuário se encontra determina os possíveis novos estados para qual ele pode se deslocar. Incluindo um número mínimo de estados, geralmente sendo eles: **atacar, recuar, explorar e procurar/pegar itens**.

Também é interessante notar que a IA mantém dois estados, um estado ideal e um estado atual. Dessa forma, é fácil para o código do jogo fornecer metas aos atores e fazê-los descobrir a melhor maneira de alcançá-los. É uma combinação interessante de uma máquina de estados com um sistema orientado por objetivos.

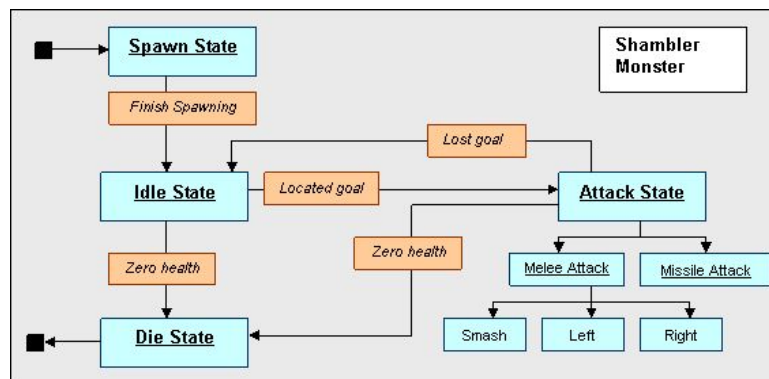


Figura 1 – Representação da máquina de estado de um inimigo.

Vale ressaltar que em Half-Life está presente a técnica chamada “máquina de estado orientada por programação”, que é outra maneira de dizer que é dependente do estado e dirigida por respostas. Ao colocar em camadas os FSMs, os desenvolvedores conseguiram alcançar uma complexidade notável na forma como o IA se comporta - monstros podem entrar em pânico quando fogem, se movimentam quando atacam o jogador e buscam reforços se perceberem que estão perdendo uma batalha. Embora seja uma técnica muito boa e altamente utilizada até os dias atuais, uma das maiores reclamações dos jogadores é o jogo se tornar previsível.

### 2.1.1 FUNCIONAMENTO DE UMA FSM

Máquinas de Estado finito são conjuntos de regras e estados que definem a situação em que o personagem se encontra e quando ela irá se alterar, ou seja, de acordo com o estado que a Máquina se encontra é como ela deve se comportar durante o jogo. Para desenvolver uma Máquina de Estados é relativamente simples e o ideal é sempre dividir seu comportamento em diversos estados controlando a transição entre eles, toda transição irá se basear nos fatores de entrada, sendo eles causados pelo jogador ou pelo ambiente.

- **Estados:** Representam uma posição no tempo que, conseqüentemente irá influenciar no comportamento do agente.
- **Transições:** São ligações entre os estados.
- **Eventos:** São ações que ocorrem ao redor do agente.
- **Condições:** São regras que devem ser preenchidas para que ocorra a mudança de estado

< <https://docplayer.com.br/8471717-inteligencia-artificial-aplicada-aos-jogos-eletronicos.html> >

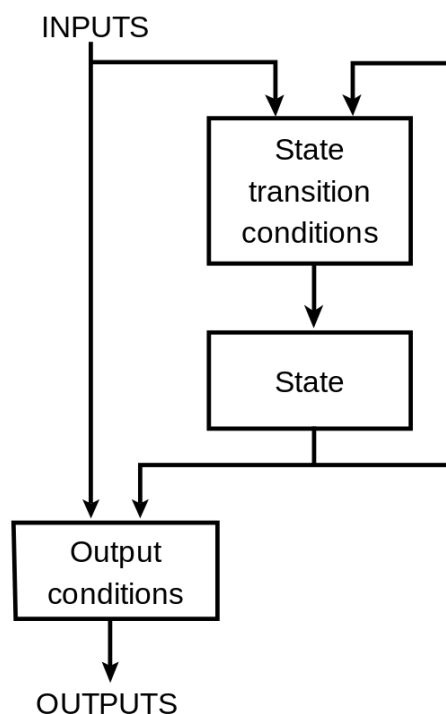


Figura 2 - Representação de entrada e saída de FSM

### 2.1.2 ESTRUTURA DE CÓDIGO

A maior parte da lógica do jogo está no subdiretório do jogo **/dll**, onde encontrará todos os arquivos necessários para construir o **hl.dll**, que também é uma estrutura para mods. Este diretório também contém o código da IA do game, que vem em uma variedade de arquivos, incluindo os nomes *\*monster\*.h*, *\*ai\*.h*, *\*ai\*.cpp* e outros arquivos miscelâneos.

O código usado na Valve é originalmente derivado do mecanismo Quake do Id, que ainda é óbvio, embora o código tenha sido convertido em C++; arquivos e estruturas são nomeados de forma semelhante.

#### Alguns aspectos da estrutura:

- **Agendador e Sistema de Metas:** Na programação de arquivos [h, cpp], você encontrará um sistema muito simples orientado por objetivos. Consiste em várias camadas de tarefas que podem ser combinadas processualmente.
- **Tarefas:** Tarefas são comportamentos atômicos curtos que são definidos para um propósito específico. Por exemplo, a maioria dos atores no Half-Life suporta o seguinte: `TASK_WALK_PATH`, `TASK_CROUCH`, `TASK_STAND`, `TASK_GUARD`, `TASK_STEP_FORWARD`, `TASK_DODGE_RIGHT`, `TASK_FIND_COVER_FROM_ENEMY`, `TASK_EAT`, `TASK_STOP_MOVING`, `TASK_TURN_LEFT`, `TASK_REMEMBER`. Eles são definidos como enumerações no arquivo de cabeçalho e implementados como métodos C++.
- **Condições:** Condições são usadas para expressar a situação de um ator no mundo. Como tudo é codificado, as condições podem ser expressas muito compactamente como um bitfield, mas nesse caso ele limita as diferentes condições a 32. Por exemplo, as condições são `COND_NO_AMMO_LOADED`, `COND_SEE_HATE`, `COND_SEE_FEAR`, `COND_SEE_DISLIKE`, `COND_ENEMY_OCCLUDED`, `COND_ENEMY_TOO_FAR`, `COND_HEAVY_DAMAGE`, `COND_CAN_MELEE_ATTACK2`, `COND_ENEMY_FACING_ME`.
- **Agendamentos:** Um cronograma é composto de uma série de tarefas (com parâmetros arbitrários) e recebe um campo de bits de condições para ajudar a especificar quando esse cronograma é inválido. O objeto de agendamento básico também tem um nome para ajudar na depuração.
- **Objetivos:** Em um nível mais alto, as metas são compostas por vários agendamentos. A lógica na meta pode selecionar uma programação conforme necessário com base em qual tarefa falhou e qual é o contexto atual. Os gols no Half-Life incluem `GOAL_ATTACK_ENEMY`, `GOAL_MOVE`, `GOAL_TAKE_COVER`, `GOAL_MOVE_TARGET` e `GOAL_EAT`.

### 2.1.3 MÁQUINA DE ESTADOS EM HALF-LIFE

Essa técnica está presente em todo o jogo praticamente, conforme demonstração posterior de um trecho do código-fonte. Além disso podemos citar algumas situações onde ocorrem a interferência da técnica, basicamente devem existir funções pelo código que validam o estado e o que está acontecendo para decidir o que virá a seguir:

- As cenas são construídas através do “disparo” de algumas ações determinadas e contam com a participação do jogador (Não são cutscenes - uma sequência sobre a qual o jogador tem nenhum ou pouco controle, interrompendo a jogabilidade e sendo



usada para avançar o enredo, reforçar o desenvolvimento do personagem principal, introduzir personagens inimigos).

- Os Guardas e Cientistas vêm até o seu personagem quando você é avistado para passar informações.
- Cientistas entram em pânico ao encontrar corpos e sangue pelo ambiente.

#### 2.1.3.1 PISCAR OLHOS E LOCALIZAR PLAYER

```
// RunAI - substituído por bullsquid
// precisa ser verificado a cada pensamento/ação.
//=====================================================
void CBullsquid :: RunAI ( void )
{
    // primeiro, fazer coisas de classe base
    CBaseMonster :: RunAI();

    if ( pev->skin != 0 )
    {
        // fechar o olho se estiver aberto.
        pev->skin = 0;
    }

    if ( RANDOM_LONG(0,39) == 0 )
    {
        pev->skin = 1;
    }

    // se não houver inimigos e estiver correndo
    if ( m_hEnemy != NULL && m_Activity == ACT_RUN )
    {
        // perseguir o inimigo. Sprint para o último bit
        if ( (pev->origin - m_hEnemy->pev->origin).Length2D() <
SQUID_SPRINT_DIST )
        {
            pev->framerate = 1.25;
        }
    }
}
```

#### 2.1.3.2 DISTÂNCIA DO ALVO E DISPARO DE SONS (GRITOS)

```
void CScientist :: RunTask( Task_t *pTask )
{
    switch ( pTask->iTask )
    {
        case TASK_RUN_PATH_SCARED:
            if ( MovementIsComplete() )
```

```
        TaskComplete();
    if ( RANDOM_LONG(0,31) < 8 )
        Scream();
    break;
case TASK_HEAL:
    if ( m_fSequenceFinished )
    {
        TaskComplete();
    }
    else
    {
        if ( TargetDistance() > 90 ) // distância do alvo
            TaskComplete();
        pev->ideal_yaw = UTIL_VecToYaw( /* ... */ );
        ChangeYaw( pev->yaw_speed );
    }
    break;

default:
    CTalkMonster::RunTask( pTask );
    break;
}
```

### 3 CONCLUSÃO

Em uma época onde os games se limitavam a copiar as fórmulas de Doom e Quake, Half-Life inovou, mesclando elementos de exploração, aventura e até mesmo terror, com os marcantes alienígenas, proporcionando um ar sombrio em dados momentos da aventura. Mesmo que inicialmente tenha sido baseado em Quake e inspirado em Doom, a dinâmica e as técnicas aplicadas no game, bem como a reescrita de seu código o tornam diferenciado. Segundo análise do site GameHall do Portal Uol, essas abordagens, inovadoras para época, abriram portas para que excelentes games como Borderlands 2, Bioshock, Far Cry 3 entre outros entrassem em cena, mostrando que o gênero de FPS poderia trazer tramas e jogos muito mais profundos e criativos, com aventuras mais dinâmicas e diversificadas do que era visto na velha e defasada fórmula criada por Doom e aprimorada com Quake, que mesmo sendo divertida já mostrava claros sinais de velhice.

Os cenários, além de diversificados, contavam com texturas muito bem feitas, vários detalhes como xícaras de café, lápis, canetas, extintores de incêndio, tudo para deixar os cenários ricos e mais realistas, sem contar que o jogador podia interagir com alguns, como apagar as luzes de alguns laboratórios no início do game e ver a reação dos personagens, algo nunca antes visto em um game até então. As armas seguiam o alto nível de qualidade, com excelentes modelagens e um cuidado especial, o que é compreensível, já que fica a mostra o tempo todo na tela, e sem deixar de lado os efeitos diversos, como explosões, raios, água, luz, sombra, entre muitos outros.

As técnicas de IA utilizadas no game, mesmo que simples possibilitaram um certo dinamismo e ‘personalidade’ aos inimigos. Máquinas de Estado Finito foram utilizadas em praticamente todas as abordagens, a interação do player ao longo do jogo faz com que existam diversas possibilidades e mesmo que com regras limitadas, ainda assim seria necessário esgotar todas as combinações/estados possíveis para o jogo se tornar repetitivo.

## 4 REFERÊNCIAS

PUC-RIO, **Inteligência Artificial em Jogos.** Disponível em: <[https://www.maxwell.vrac.puc-rio.br/7861/7861\\_3.PDF](https://www.maxwell.vrac.puc-rio.br/7861/7861_3.PDF)>. Acesso em: 06 outubro. 2018.

NEOGAMER, **Unseen - O desenvolvimento de Half-Life.** Disponível em: <<http://www.neogamer.com.br/2011/06/unseen-o-desenvolvimento-de-half-life.html>>. Acesso em: 09 outubro. 2018.

AI GAME DEV. **The ai from half-life's sdk in retrospective.** Disponível em: <<http://aigamedev.com/open/article/half-life-sdk/>>. Acesso em: 06 out. 2018.

AI-DEPOT. **A practical analysis of fsm within the domain of first-person shooter (fps) computer game.** Disponível em: <<http://ai-depot.com/FiniteStateMachines/FSM-Practical.html>>. Acesso em: 09 out. 2018.

DOCPLAYER. **Inteligência artificial aplicada aos jogos eletrônicos.** Disponível em: <<https://docplayer.com.br/8471717-inteligencia-artificial-aplicada-aos-jogos-eletronicos.html>>. Acesso em: 09 out. 2018.

GAMASUTRA. **Game ai: the state of the industry.** Disponível em: <[https://www.gamasutra.com/view/feature/131705/game\\_ai\\_the\\_state\\_of\\_the\\_industry.php](https://www.gamasutra.com/view/feature/131705/game_ai_the_state_of_the_industry.php)>. Acesso em: 09 out. 2018.

TECMUNDO. **O que é inteligência artificial?.** Disponível em: <<https://www.tecmundo.com.br/intel/1039-o-que-e-inteligencia-artificial-.htm>>. Acesso em: 08 out. 2018.

THE ELECTRICAL ENGINEERING AND COMPUTER SCIENCE (EECS). **Ai for animation and combat.** Disponível em: <<https://web.eecs.umich.edu/~sugih/courses/eecs494/fall06/lectures/lecture17-fps.pdf>>. Acesso em: 08 out. 2018.

GAMEHALL. **Análise de half-life.** Disponível em: <<http://gamehall.uol.com.br/v10/half-life/>>. Acesso em: 09 out. 2018.

## LISTA DE ILUSTRAÇÕES

**Figura 1 – Representação da máquina de estado de um inimigo:**

<<http://ai-depot.com/FiniteStateMachines/FSM-Practical.html>>

**Figura 2 - Representação de entrada e saída de FSM:**

<[https://pt.wikipedia.org/wiki/M%C3%A1quina\\_de\\_estados\\_finita#/media/File:Finite\\_State\\_Machine\\_Logic.svg](https://pt.wikipedia.org/wiki/M%C3%A1quina_de_estados_finita#/media/File:Finite_State_Machine_Logic.svg)>