



**REALIDADE VIRTUAL E INTELIGÊNCIA
ARTIFICIAL**

INTELIGÊNCIA ARTIFICIAL, INTRODUÇÃO

Paulo Salvatore

RESUMO

A inteligência artificial é um ramo das ciências da computação que vem ganhando cada vez mais força no mercado. Nos jogos isso não é diferente, um bom uso da I.A. garante a melhor jogabilidade e mantém os jogadores mais entretidos e desafiados. Nesse capítulo entenderemos de onde surgiu o conceito de I.A., sua evolução e como isso influencia os games desde então, conhecendo os principais tipos de análise e visualizando como todo esse ecossistema influencia desenvolvedores e jogadores.

Palavras-chave: Inteligência Artificial; Machine Learning; Algoritmos de Busca.

LISTA DE FIGURAS

Figura 1.1 – Fantasmas do Pac-Man apresentavam uma IA acima dos jogos da época.	5
Figura 1.2 – Campos da Inteligência Artificial.	6
Figura 1.3 – Machine Learning vs Deep Learning.....	7
Figura 1.4 – Estudo de Deep Learning no Mario.....	8
Figura 1.5 – Spacewar!.	9
Figura 1.6 – Pong.....	10
Figura 1.7 – Gun Fight – IA com padrões de movimento.	10
Figura 1.8 – Exemplo de FSM.....	12
Figura 1.9 – Fuzzy Logic.	13
Figura 1.10 – Exemplo de Flock no Age of Empires.	14
Figura 1.11 – Exemplo de Árvore de Decisão em Jogos.	15
Figura 1.12 –Visão geral das redes neurais no aprendizado profundo.	16
Figura 1.13 – Estrutura básica dos algoritmos evolutivos.	17
Figura 1.14 – Possibilidades de caminho entre cidades no Jogo Ragnarök Online podem ser facilmente calculados por algoritmos de busca.	18
Figura 1.15 – Exemplo de grafo calculado pelo BFS.	19
Figura 1.16 – Exemplo de grafo calculado pelo DFS.	20
Figura 1.17 – Grafo calculado pelo algoritmo de dijkstra.	21
Figura 1.18 – Exemplo de algoritmo A* aplicado em um ambiente 3D.	21
Figura 1.19 – Algoritmo A* aplicado em um grid.	22
Figura 1.20 – Algoritmo A* é aplicado em jogos como o Ragnarök Online, cuja movimentação é baseada em uma matriz 2x2.	23
Figura 1.21 – O jogo ScareThemOff! Utiliza um sistema de waypoints para movimentação dos NPCs.	24
Figura 1.22 – Unity’s NavMesh.	25
Figura 1.23 – Exemplo de um NavMesh complexo aplicado em um cenário com dois andares.	26
Figura 1.24 – Personagens de I.A. do jogo Pac-Man.....	27
Figura 1.25 – Cubo Mágico (Rubik’s Cube).....	28
Figura 1.26 – Ser Humano Resolvendo um Cubo Mágico.	29
Figura 1.27 – IA Resolvendo um Cubo Mágico.....	29
Figura 1.28 – Pôquer.....	30
Figura 1.29 – Garry Kasparov vs Deep Blue.	31
Figura 1.30 – Braço robótico jogando xadrez contra garoto.....	32
Figura 1.31 – Watson contra Ken e Brad em Jeopardy!.....	33
Figura 1.32 – AlphaGo contra Lee Se-dol no jogo chinês Go!.	34
Figura 1.33 – Team Human na partida contra a máquina da OpenAI	35
Figura 1.34 – Especificações técnicas do processamento utilizado para os cálculos da IA.....	36

SUMÁRIO

6	INTELIGÊNCIA ARTIFICIAL, INTRODUÇÃO	5
6.1	Introdução.....	5
6.2	Inteligência Artificial nos Games.....	8
6.2.1	Técnicas de Inteligência Artificial	11
6.2.2	Algoritmos de Busca	17
6.2.3	Exemplos de IA em Jogos Eletrônicos	26
6.2.4	IA: Derrotando Seres Humanos	27
7	REFERÊNCIAS	37

6 INTELIGÊNCIA ARTIFICIAL, INTRODUÇÃO

6.1 INTRODUÇÃO

Antes de entendermos como aplicar inteligência artificial aos games, precisamos entender de uma forma geral o que ela realmente é e como podemos aplicá-la. O termo, que foi criado em 1956, vem ganhando cada vez mais força no cenário tecnológico graças as suas mais diversas aplicações, sendo sua aplicação no ambiente corporativo a que recebe mais avanços atualmente. Dentro da inteligência artificial, existem outros campos chamados Machine Learning e Deep Learning, que representam conceitos específicos que buscam resolver problemas no mundo real.

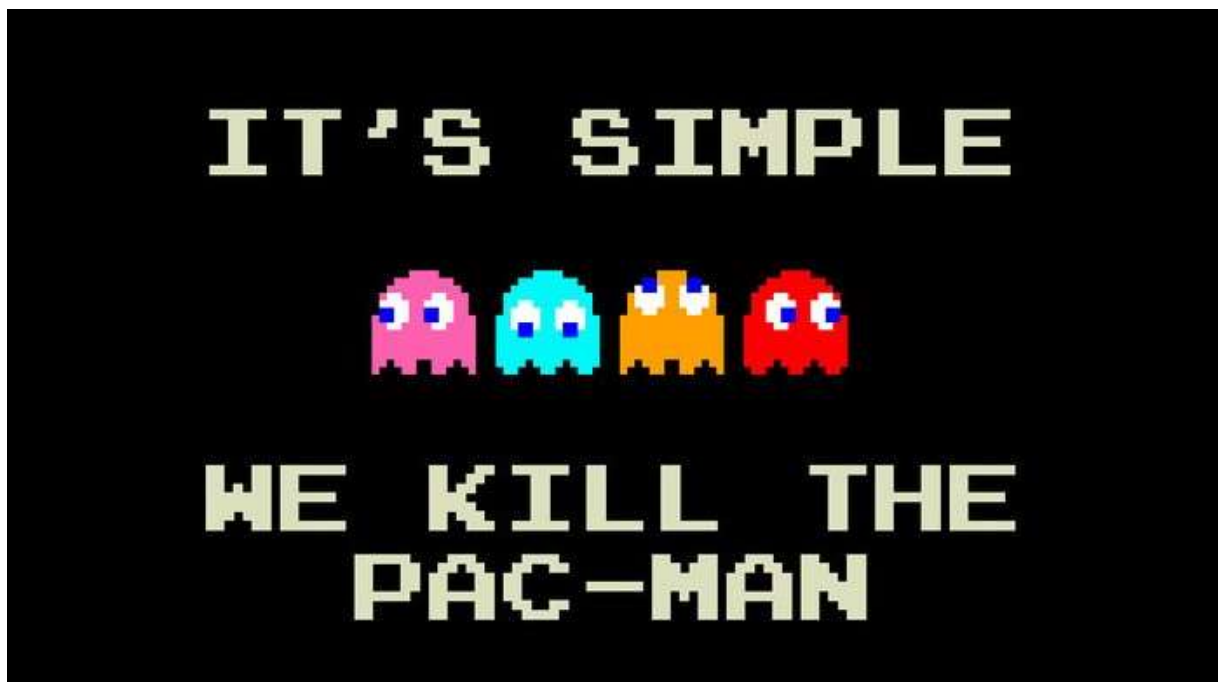


Figura **Erro! Fonte de referência não encontrada..1** – Fantasmas do Pac-Man apresentavam uma IA acima dos jogos da época.
Fonte: Tech Tudo (2014).

A **Inteligência Artificial (IA)** é a capacidade do computador de reproduzir uma inteligência similar à humana. Aplicando isso em uma linguagem de programação, qualquer rotina de condições (ifs e elses) pode ser considerada uma inteligência artificial no que diz respeito à tomada de decisão por parte de uma IA, por exemplo, em relação a informações recebidas do ambiente.

Exemplo: Sempre que o jogador entrar na sala, a IA vai identificar essa movimentação e vai passar a perseguir o jogador. Para realização dessa tarefa, precisamos encontrar uma forma de detectar se o jogador entrou na sala, colocando um trigger na entrada da sala que avisa a IA ou detectando pela proximidade do jogador em relação à IA.

Resumidamente, qualquer comportamento que reproduza à inteligência humana e sua capacidade de tomada de decisões, identificação de comportamentos, reconhecimento de imagens é considerado inteligência artificial, independente da maneira que isso é realizado.

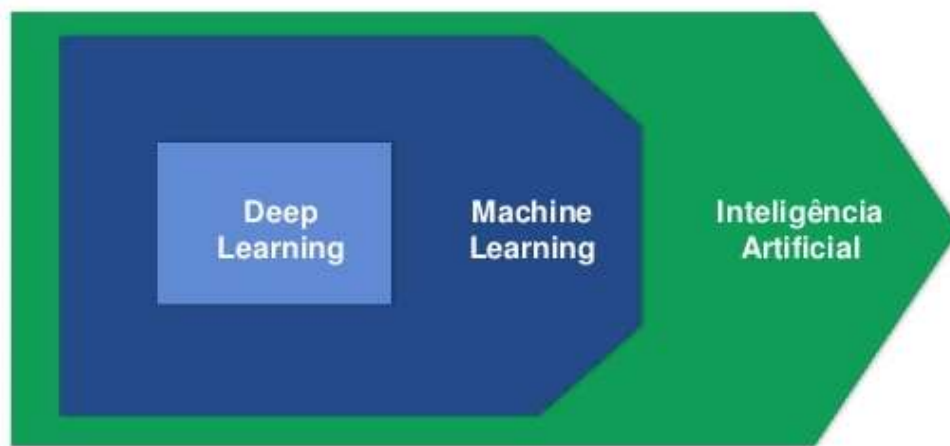


Figura **Erro! Fonte de referência não encontrada..2** – Campos da Inteligência Artificial.
Fonte: Guilherme Campos no GDG Curitiba (2017).

Aprofundando um pouco no estudo à inteligência artificial, temos outras duas classificações: **Machine Learning** e **Deep Learning**. Ambas consistem em campos da inteligência artificial com implementações distintas que buscam resolver problemas específicos.

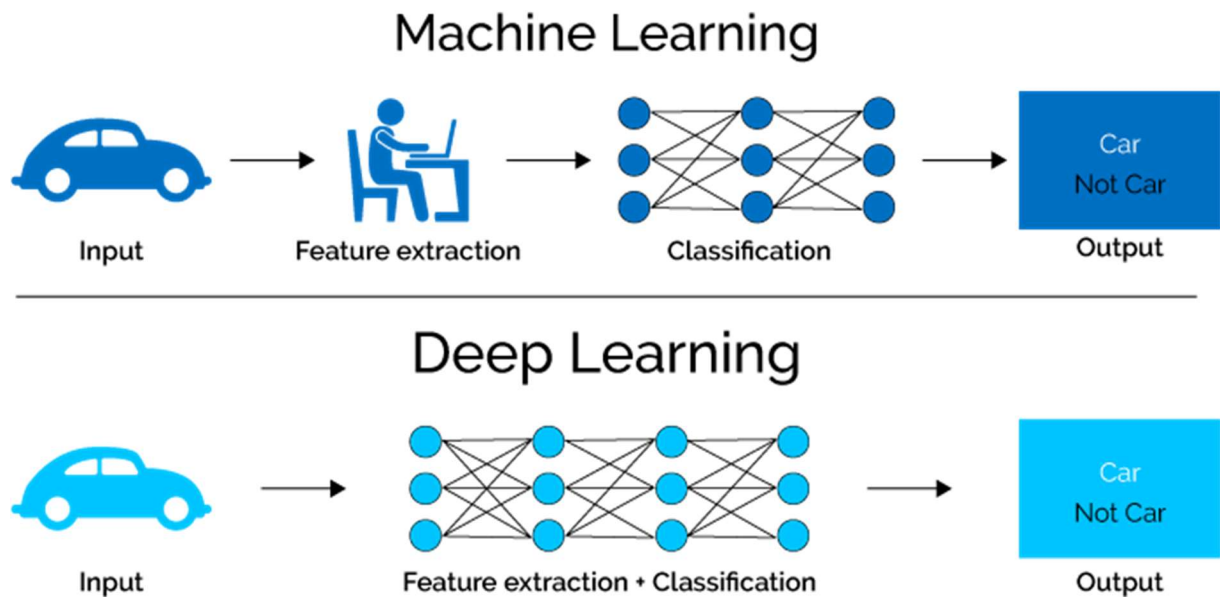


Figura **Erro! Fonte de referência não encontrada..3** – Machine Learning vs Deep Learning.
Fonte: George Seif no Medium (2018).

O **Machine Learning**, também conhecido como **Aprendizagem de Máquina**, é o campo de estudo que fornece aos computadores a capacidade de aprender sem serem explicitamente programados, aprendendo com seus erros e através de uma série de dados de utilização que são fundamentais para o aprendizado. Exemplos de aplicações incluem reconhecimento ótico de caracteres (OCR), processamento de linguagem natural, reconhecimento de fala, reconhecimento de escrita, visão computacional, locomoção de robôs, entre outros.

O **Deep Learning**, também conhecido como **Aprendizagem Profunda**, é um ramo do Machine Learning que utiliza algoritmos matemáticos conhecidos como redes neurais e redes convolucionais para aplicações que já são feitas no machine learning, porém, com essas novas técnicas é possível obter resultados cada vez mais satisfatório. Esse termo, apesar de estar presente desde a década de 1960, está ganhando mais destaque na última década, pois somente com o avanço do poder computacional foi possível obter resultados cada vez mais satisfatórios em um menor espaço de tempo.

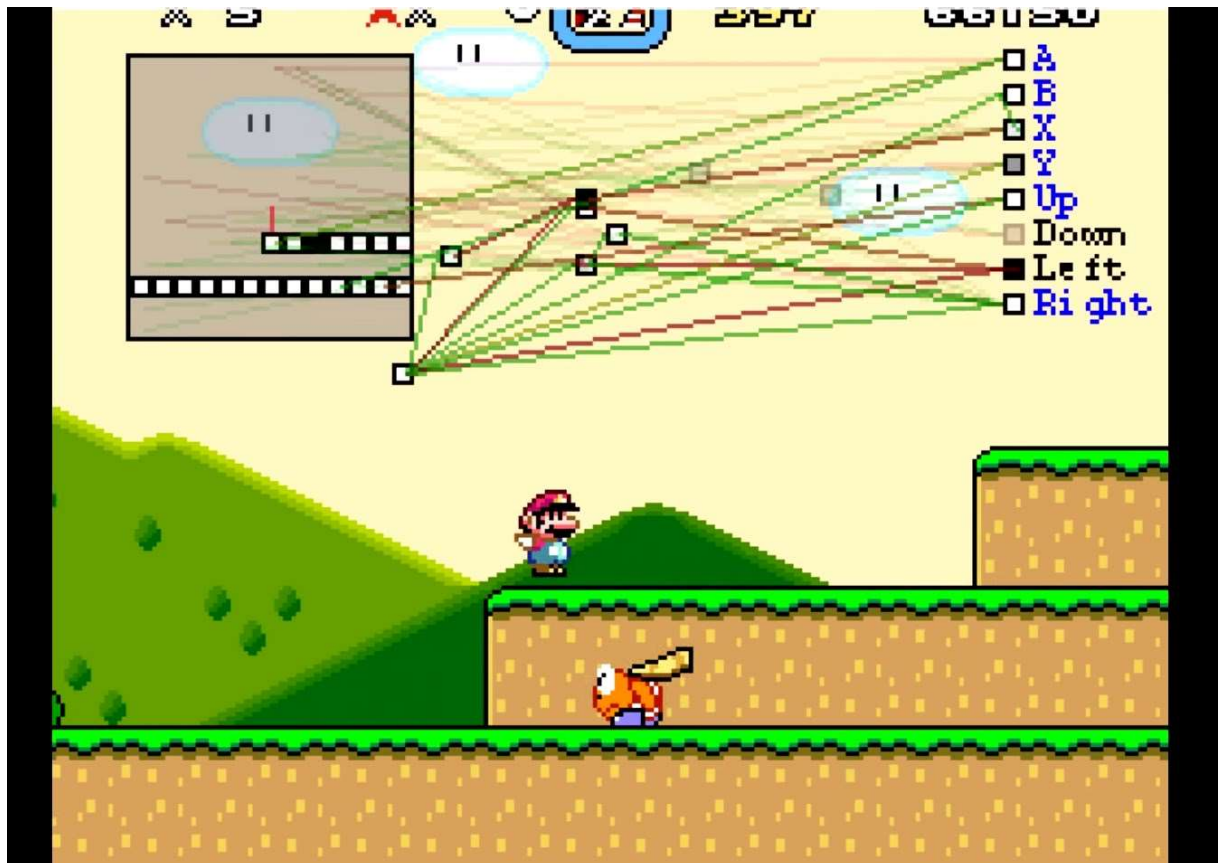


Figura **Erro! Fonte de referência não encontrada..4** – Estudo de Deep Learning no Mario.
Fonte: Aleju/MarioAI (2016).

Com isso temos uma introdução do que é inteligência artificial nos dias de hoje, cada vez mais aplicada no nosso dia-a-dia, contribuindo em todas as áreas da computação. Fazendo um paralelo com o mundo dos jogos, inicialmente temos uma aplicação muito superficial da inteligência artificial, porém, cada vez mais jogos da atualidade se preocupam com esse tema e passam a inserir fundamentos de aprendizagem de máquina a fim de melhorar a jogabilidade e tornar os jogos mais dinâmicos e interessantes.

6.2 INTELIGÊNCIA ARTIFICIAL NOS GAMES

Um dos primeiros jogos eletrônicos de computador, o Spacewar!, de 1961, era bem simples e possuía dois personagens que deveriam combater entre si. Existem vários jogos desse estilo e parece bem óbvio como seriam as opções para esse jogo: “Um Jogador”, jogando contra a CPU; e “Dois Jogadores”, um jogador contra o outro. Entretanto, o Spacewar! era jogado apenas por dois jogadores, sendo que cada jogador deve controlar um personagem.

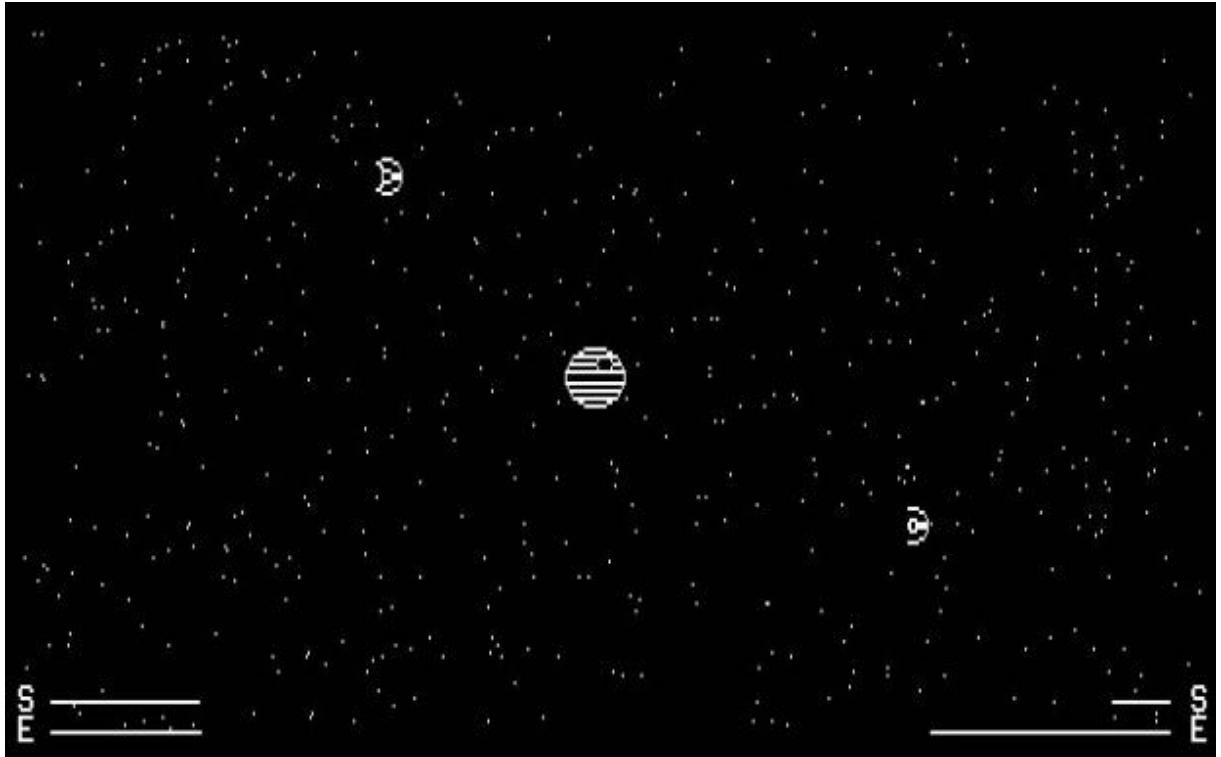


Figura **Erro! Fonte de referência não encontrada..5** – Spacewar!.
Fonte: MIT (1961).

Outro jogo até mais conhecido é o Pong, lançado pela Atari em 1972, que consistia em dois jogadores rebatendo uma bola em um campo semelhante a uma quadra de tênis. Mesmo com mais de 10 anos após o lançamento do Spacewar! e com estudos do campo de inteligência artificial, a aplicação nos games ainda não era tão óbvia assim.

As primeiras tentativas para fornecer uma opção do jogador usufruir do game sozinho foi inserindo movimentações aleatórias nos personagens, o que dificultava com que o jogador conseguisse prever o comportamento da máquina e não proporcionava um nível de dificuldade que desafiasse os jogadores. Essa implementação pode ser chamada de “hard-coded AI”, ou seja, uma inteligência artificial onde todo o seu comportamento é escrito diretamente via código

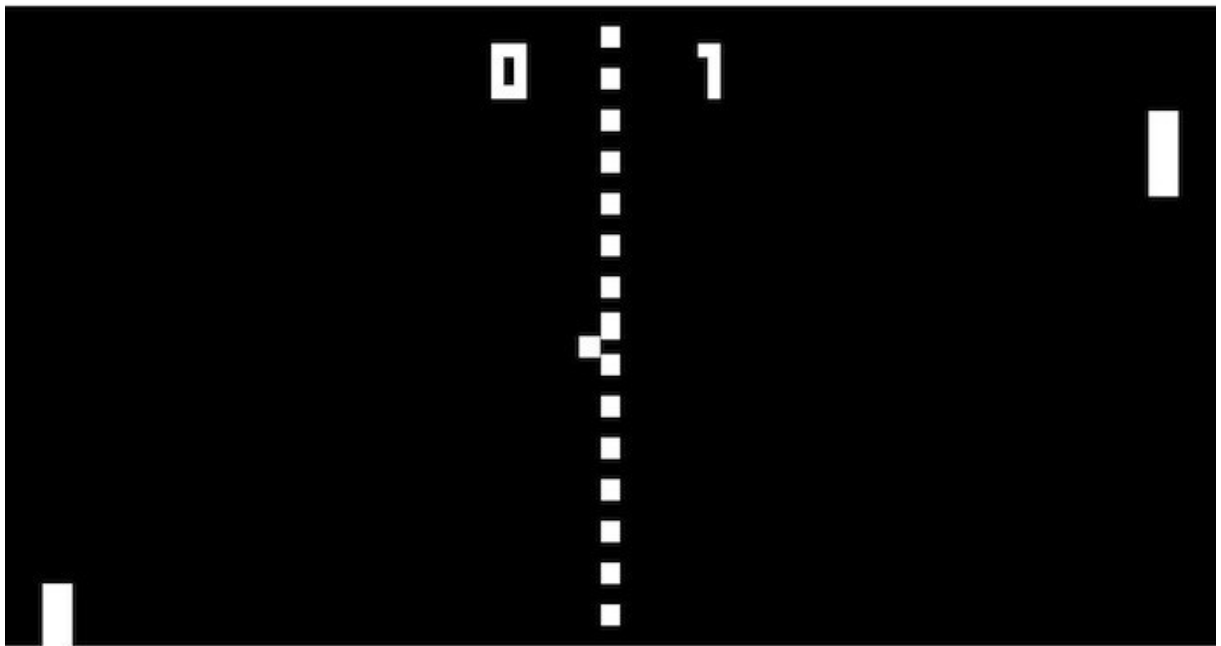


Figura **Erro! Fonte de referência não encontrada..6** – Pong.
Fonte: Atari (1972).

Com o avanço das criações, jogos como Pursuit (Atari, 1974) e Gun Fight (Atari, 1975) passaram a incluir inteligência artificial hard-coded, em que os programadores escreveram rotinas de padrões de movimento para os personagens controlados pela CPU.



Figura **Erro! Fonte de referência não encontrada..7** – Gun Fight – IA com padrões de movimento.
Fonte: Atari (1975).

Space Invaders, Donkey Kong e todos esses jogos dessa geração rodavam com regras bem simples e ações definidas previamente. Os personagens controlados por IA não possuíam a habilidade de tomar decisões. Algumas vezes as decisões eram definidas via script e executadas de forma aleatória para que o comportamento dos agentes parecesse mais imprevisível.

6.2.1 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Em busca de uma jogabilidade mais interessante e desafiadora, os jogos passaram a estudar diferentes algoritmos de inteligência artificial e a aplicar técnicas distintas que cumprissem com esse objetivo. Entre essas técnicas estão: **Máquina de estado finita (FSM)**, **Fuzzy**, **Flocking**, **Árvores de decisão** e **Redes Neurais**.

6.2.1.1 MÁQUINA DE ESTADO FINITA

Máquina de estado finita (FSM – Finite State Machine) é uma das técnicas de IA mais utilizada e consiste em um modelo com um número finito de estados, sendo que a máquina está em apenas um desses estados por vez, chamado de estado atual. Um estado armazena informações sobre estados anteriores, refletindo as mudanças desde a entrada num estado. Uma transição indica uma mudança entre estados e ocorre após uma determinada condição ser realizada. Quando a máquina está em determinado estado, ela passa a executar ações que descrevem as atividades a serem realizadas nesse determinado momento, além de sempre verificar se alguma das condições de mudança de estados são atendidas.

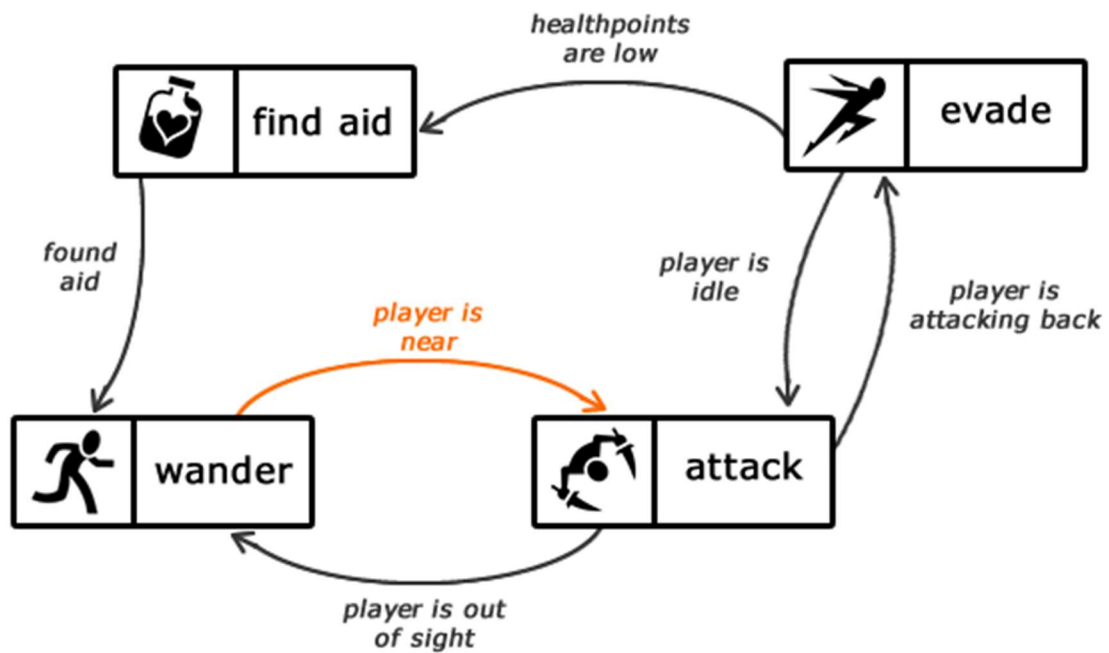


Figura **Erro! Fonte de referência não encontrada..8** – Exemplo de FSM.
Fonte: TutsPlus (2013).

6.2.1.2 LÓGICA FUZZY

A **lógica fuzzy**, também conhecida como lógica difusa ou nebulosa, é baseada no princípio da incerteza. Diferente da lógica booleana onde a resposta é sempre “sim” ou “não”, o objetivo da lógica fuzzy é responder algo como “talvez” ou “quase” quando for aplicável. A aplicação dessa lógica torna possível a produção de cálculos com informações imprecisas, assim como os seres humanos fazem

Desse modo é possível avaliar conceitos não-quantificáveis, como temperatura (quente, morno, médio, etc...), felicidade (radiante, feliz, apático, triste, etc...), entre outros. Definir os atributos em diferentes situações pode ajudar a um controlador fuzzy determinar qual a hora de atacar ou fugir e até mesmo ajuda-lo a definir qual arma mais eficiente a ser utilizada dependendo da situação.

Também é possível aplicar a lógica fuzzy em conjunto com as máquinas de estado finitas, uma prática que é bastante comum. Uma das possíveis aplicações dessa técnica é calcular o “nível de medo” de um personagem em relação a um jogador.

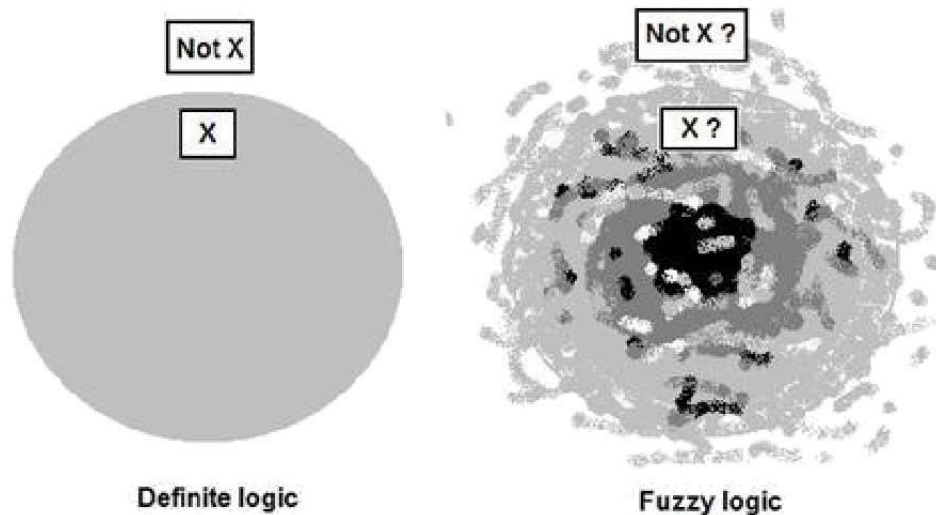


Figura **Erro! Fonte de referência não encontrada..9** – Fuzzy Logic.
Fonte: Michele Pirovano (2012).

6.2.1.3 FLOCKING

O comportamento **Flocking** é um estudo feito a partir da análise comportamental de um grupo de indivíduos (traduzido do inglês *Flock*) como o de um cardume (peixes) ou de um bando (pássaros). Ao aplicar esse estudo é possível simular ambientes semelhante com uma alta fidelidade e um alto grau de realismo.

Essa técnica uma alternativa à construção de scripts para uma grande quantidade de indivíduos, transformando a movimentação em uma percepção do personagem em relação ao meio ambiente. A cada novo ciclo de análise há uma atualização da percepção das redondezas pelo personagem, que realiza uma ação através disso.

Embora a técnica seja um tanto quanto limitada, sua implementação é simples e eficiente. Um grande exemplo de demonstração dessa técnica são jogos de primeira pessoa como Counter Strike e Half-life, e jogos de estratégia em tempo real como Age of Empires e Warcraft.



Figura **Erro! Fonte de referência não encontrada..10** – Exemplo de Flock no Age of Empires.
Fonte: Age of Empires II (1999).

6.2.1.4 MACHINE LEARNING

A técnica de **machine learning** é utilizada nos jogos por meios de árvores de decisão e são inseridas quando são necessários recursos como predição e classificação. Essa técnica permite prever e realizar ações baseado em dados coletados anteriormente.

Através da coleta e da análise desses dados, é possível fazer com que o computador aprenda com o jogador ou com outros personagens conforme o jogo avança. Isso cria comportamentos interessantes como quando em um jogo onde a IA procura o jogador por um cenário com vários pontos de esconderijo, na medida que o jogador passa a se esconder mais dentro de armários/guarda-roupas, a IA passa a procurar mais dentro desses locais.

As informações são organizadas em árvores de decisão, em que esses dados bagunçados são estruturados de uma maneira que facilita a compreensão e a tomada

de decisão por parte da inteligência artificial. Isso faz com que as informações sejam relevantes apenas quando aparecem com mais frequência, evitando com que ruídos atrapalhem todo o funcionamento do sistema, permitindo um aprendizado mais eficiente.

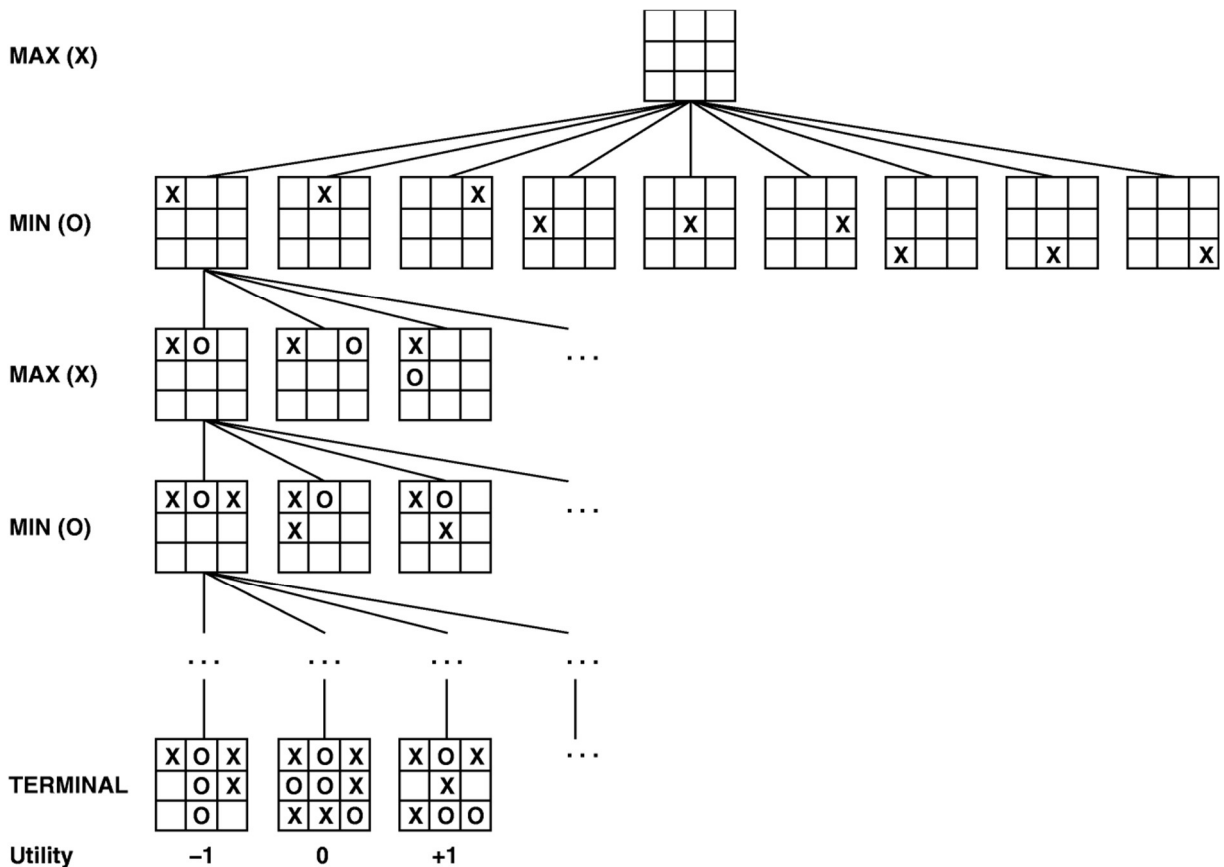


Figura **Erro! Fonte de referência não encontrada..11** – Exemplo de Árvore de Decisão em Jogos.
Fonte: Lorenzo Candiago (2017).

6.2.1.5 REDES NEURAIS

As **redes neurais** são inspiradas em sistemas biológicos, em particular o sistema nervoso central (cérebro) de um animal e são bastante aplicadas em sistemas que utilizam a tomada de decisão, o processamento de informações e a otimização de dados.

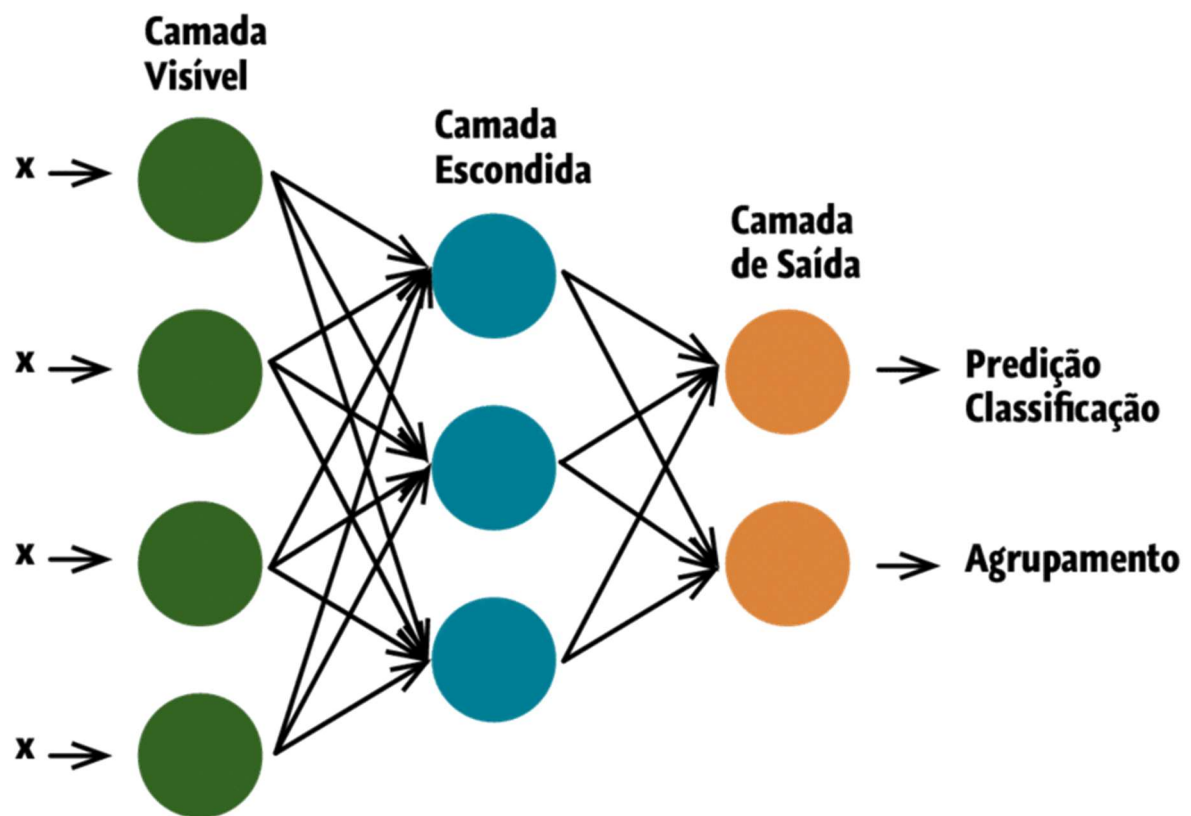


Figura **Erro! Fonte de referência não encontrada..12** –Visão geral das redes neurais no aprendizado profundo.

Fonte: Vinicius Soares da Silveira (2017).

O sistema artificial consiste em neurônios interconectados que podem computar valores de entrada que são ativados baseados nas informações recebidas. Em um exemplo com o reconhecimento de escrita manual, o conjunto de neurônios de entrada são ativados pelos pixels da imagem recebida. Os dados adquiridos são repassados, ponderados e transformados por uma função previamente programada, a outros neurônios. Esse processo é realizado diversas vezes até que uma condição de saída seja satisfatória, determinando, no caso, qual caractere foi lido.

No caso dos jogos, essa técnica é geralmente aplicada de maneira off-line, pois alterações de parâmetros podem gerar diversos comportamentos inesperados da IA fazendo com que ela perca o sentido. O principal objetivo dessa aplicação é fazer com que a inteligência artificial aprenda a imitar o comportamento do jogador.

Uma outra técnica que se assemelha bastante é a de **Algoritmos Evolutivos** (AE), também conhecidos como algoritmos genéticos, em que seu uso aplicado aos jogos se assemelha com o das redes neurais. Essa definição é um pouco mais recente e consiste em um processo de avaliação constante e de evolução genética, em que um algoritmo é gerado, testado no meio e apenas os melhores indivíduos são

selecionados para a próxima geração, que consiste no cruzamento dos algoritmos e na criação de novos indivíduos aplicando o processo de mutação aleatória. Os indivíduos gerados são testados novamente e esse processo é repetido até que obtenha um resultado satisfatório.



Figura **Erro! Fonte de referência não encontrada..13** – Estrutura básica dos algoritmos evolutivos.
Fonte: Jarryer De Martino (2015).

6.2.2 ALGORITMOS DE BUSCA

Os algoritmos de busca são uma parte importantíssima no mundo dos games, afinal, são responsáveis por resolver diversos problemas. O principal objeto do algoritmo é receber um problema e retornar uma solução depois de calcular um número possível de soluções.

“A maioria dos algoritmos estudados por cientistas da computação que resolvem problemas são algoritmos de busca.” – Aquiles Burlamaqui, UERN

Geralmente isso é feito através de alguns processos matemáticos que otimizam a análise do problema, que principalmente no caso dos jogos, essa solução precisa ser dada quase que instantaneamente.



Figura **Erro! Fonte de referência não encontrada..14** – Possibilidades de caminho entre cidades no Jogo Ragnarök Online podem ser facilmente calculados por algoritmos de busca.

Fonte: Gravity (2002).

6.2.2.1 BUSCA EM LARGURA (BREADTH-FIRST SEARCH – BFS)

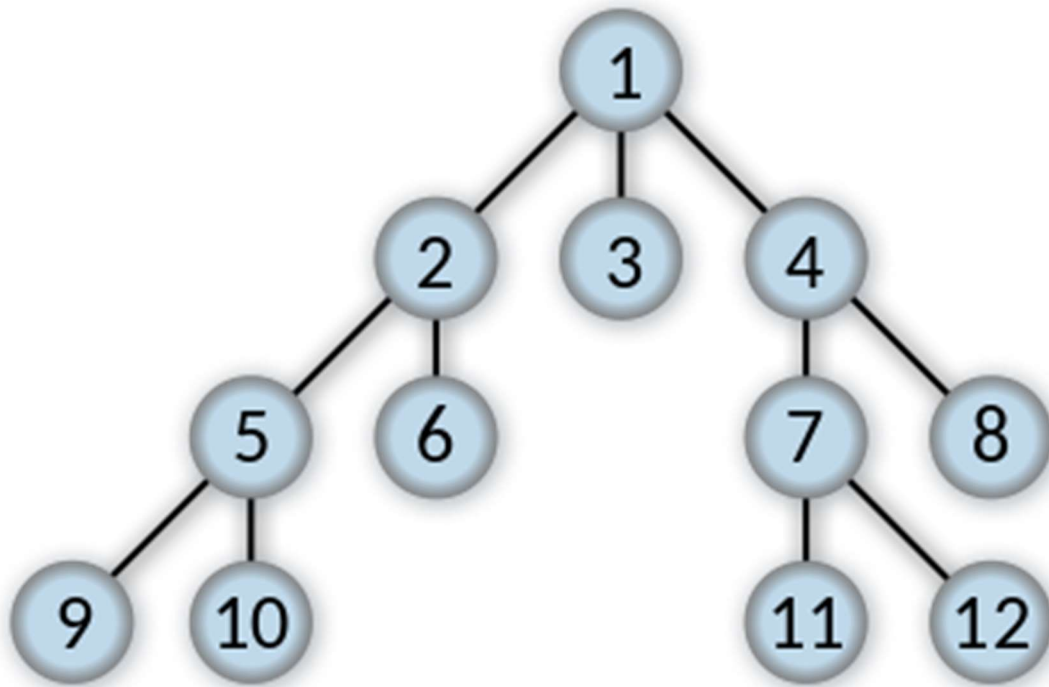


Figura **Erro! Fonte de referência não encontrada..15** – Exemplo de grafo calculado pelo BFS.
Fonte: ITAbits (desconhecido).

O algoritmo de busca em largura consiste em, a partir de um ponto de origem, representado na Figura **Erro! Fonte de referência não encontrada..15** com o número 1, expandir a classificação dos pontos pelos pontos mais próximos, numerando-os em sequência. A análise inicial é feita em por linha (“largura”), antes de descer os níveis (“profundidade”), expandindo nessa ordem até que acabem todos os pontos.

6.2.2.2 BUSCA EM PROFUNDIDADE (DEPTH-FIRST SEARCH – DFS)

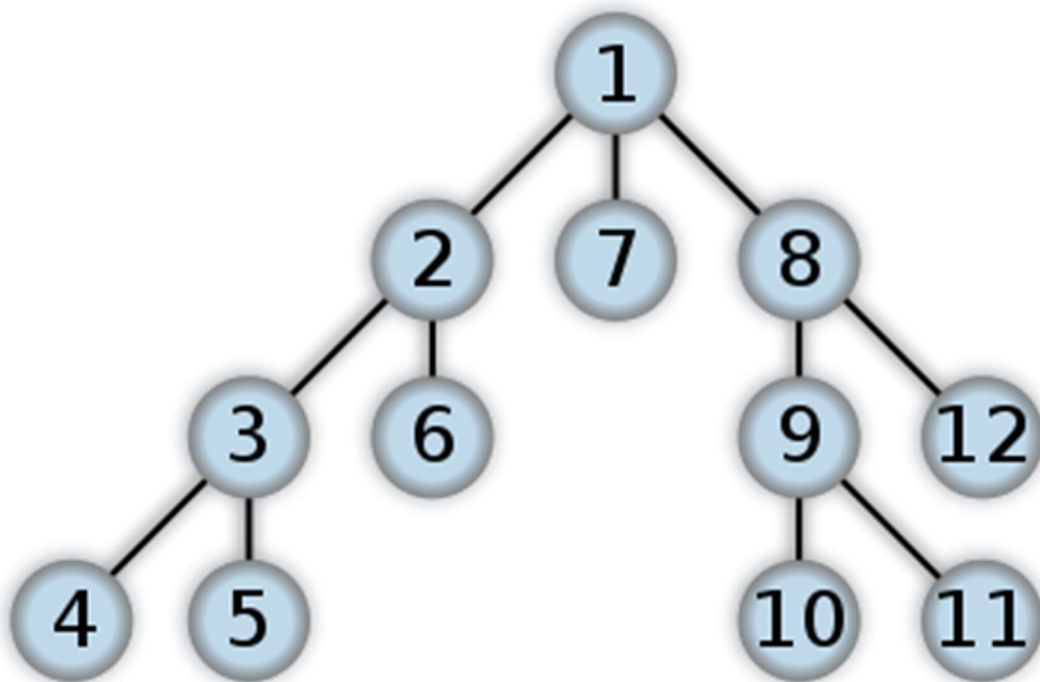


Figura **Erro! Fonte de referência não encontrada..16** – Exemplo de grafo calculado pelo DFS.
Fonte: ITAbits (desconhecido).

O algoritmo de busca em largura consiste em, a partir de um ponto de origem, representado na Figura **Erro! Fonte de referência não encontrada..16** com o número 1, expandir a classificação dos pontos pelos pontos mais próximos, numerando-os em sequência. A análise inicial é feita em por linha (“largura”), antes de descer os níveis (“profundidade”), expandindo nessa ordem até que acabem todos os pontos.

6.2.2.3 ALGORITMO DE DIJKSTRA

O algoritmo de dijkstra foi concebido em 1956 e publicado em 1959 pelo holandês Edsger Dijkstra, tem como objetivo solucionar o problema de busca de caminho, levando em consideração a distância entre os pontos e não apenas a ligação entre os pontos em si, como é possível notar na Figura **Erro! Fonte de referência não encontrada..17** – Grafo calculado pelo algoritmo de dijkstra.

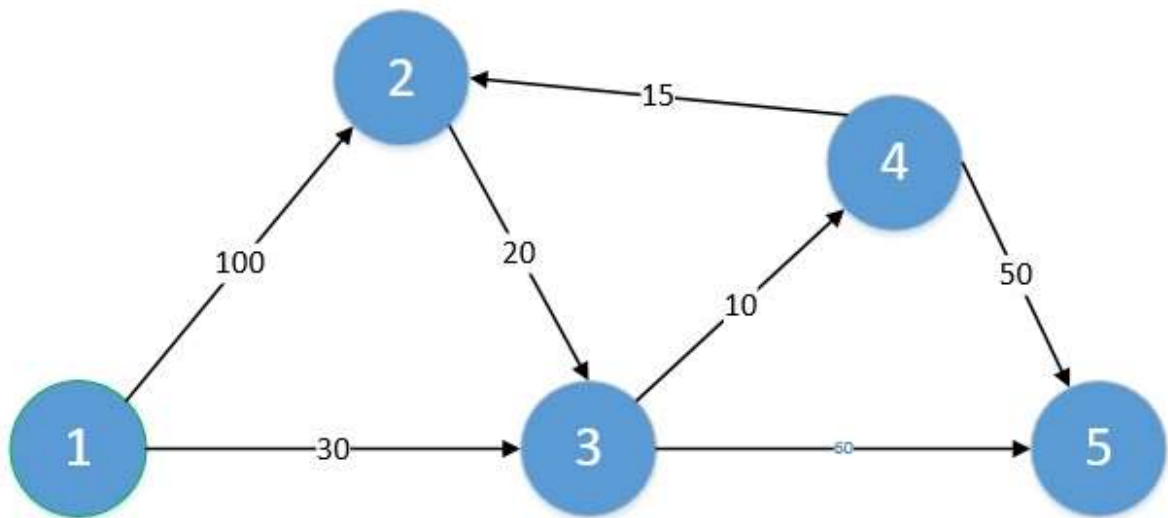


Figura **Erro! Fonte de referência não encontrada..17** – Grafo calculado pelo algoritmo de dijkstra.
Fonte: Ingenieria Industrial Online (desconhecido).

6.2.2.4 ALGORITMO A*

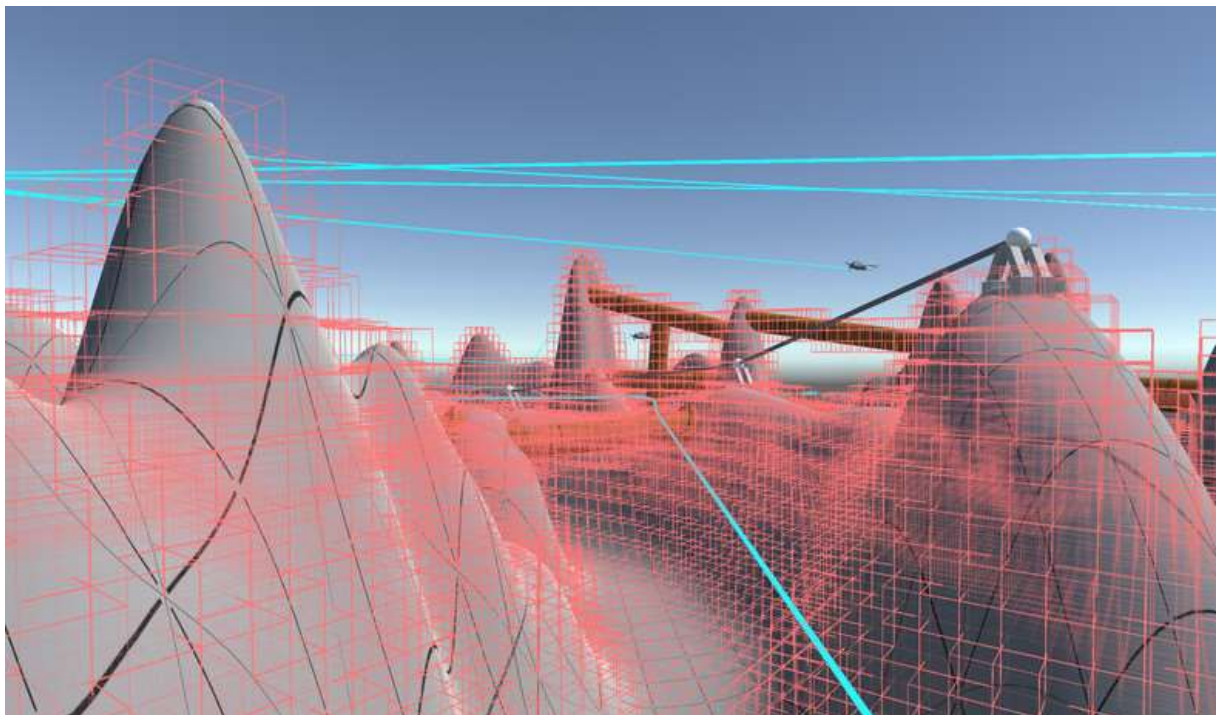


Figura **Erro! Fonte de referência não encontrada..18** – Exemplo de algoritmo A* aplicado em um ambiente 3D.

Fonte: PathFinder 3D (2017).

O Algoritmo A* é um dos algoritmos de busca de caminho mais conhecidos nos games e é combina algoritmos como o BFS (Busca em Largura) e o Algoritmo de Dijkstra, citados anteriormente. Descrito pela primeira vez em 1968, foi chamado inicialmente apenas de algoritmo A, porém, pelo fato da utilização com uma heurística apropriada ter um comportamento ótimo, passou a ser conhecido por A*.

6.2.2.4.1 Grid Based Solution

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

Figura **Erro! Fonte de referência não encontrada.**19 – Algoritmo A* aplicado em um grid.
Fonte: Wikipedia (2017).

A aplicação do algoritmo é simples, consiste em numerar todos os pontos adjacentes a partir do ponto inicial. A numeração indica quantos pontos diretos há entre o ponto inicial e o ponto atual, não considerando caminhos diagonais. Com a numeração determinada, o algoritmo busca calcular o caminho que obtém a menor soma total, sendo que é possível calcular utilizando caminhos diagonais ou sem utilizar as diagonais. Ao final, obtém-se a solução com a menor soma total, incluindo todos os pontos calculados.



Figura **Erro! Fonte de referência não encontrada..20** – Algoritmo A* é aplicado em jogos como o Ragnarök Online, cuja movimentação é baseada em uma matriz 2x2.
Fonte: Ragnarök (2002).

6.2.2.4.2 Waypoint Based Solution

Outra solução possível para análise do algoritmo A* é a utilização de pontos organizados em waypoints. Como é possível notar na Figura **Erro! Fonte de referência não encontrada..21**, o jogo ScareThemOff!, de 2016, utiliza uma organização de pontos conhecidos como waypoints, espalhados em locais estratégicos pelo cenário. Para que os NPCs se locomovam pelo cenário eles precisam decidir para onde querem se mover e o algoritmo analisa os pontos para buscar a melhor solução.



Figura **Erro! Fonte de referência não encontrada..21** – O jogo ScareThemOff! Utiliza um sistema de waypoints para movimentação dos NPCs.
Fonte: SalvatoreGames (2016).

6.2.2.5 UNITY'S NAV MESH

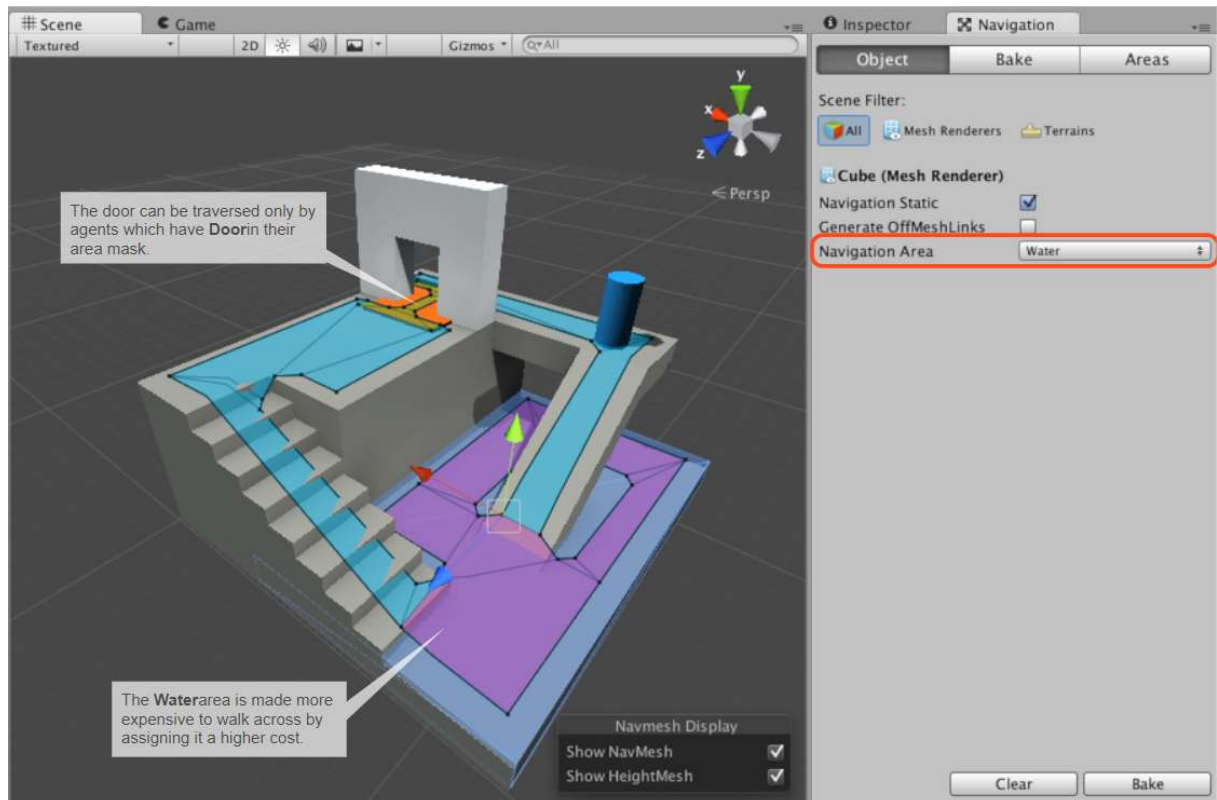


Figura **Erro! Fonte de referência não encontrada..22** – Unity's NavMesh.
Fonte: Unity (2018).

A Unity possui um sistema de navegação próprio, com uma implementação muito tranquila, mas que funciona apenas para objetos 3D. Esse sistema possui partes fundamentais para garantir seu funcionamento:

- **NavMesh (nome encurtado de Navigation Mesh):** consiste em uma malha de navegação com uma estrutura de dados que descreve a superfície navegável de um cenário, permitindo que personagens se movimentem por essa área. Essa geometria de navegação é gerada automaticamente pela Unity, desde que a superfície esteja com a opção "Navigation Static" ativada. A geração do NavMesh é feita durante o modo de edição, através de um processo conhecido como "NavMesh's Bake".
- **NavMeshAgent:** componente que auxilia na criação de personagens que entenderão um NavMesh para movimentação.
- **Off-Mesh Link:** componente que permite a inserção de atalhos de movimentação que não estão representados no NavMesh, permitindo

com que personagens pulem sobre cercas ou abrir uma porta antes de andar por ela.

- **NavMesh Obstacle:** componente que permite a criação de obstáculos que se movem pelo cenário, forçando os NavMesh Agents a evitarem-nos durante sua movimentação. Os NavMesh Obstacles influenciam em um NavMesh em tempo real durante o jogo.

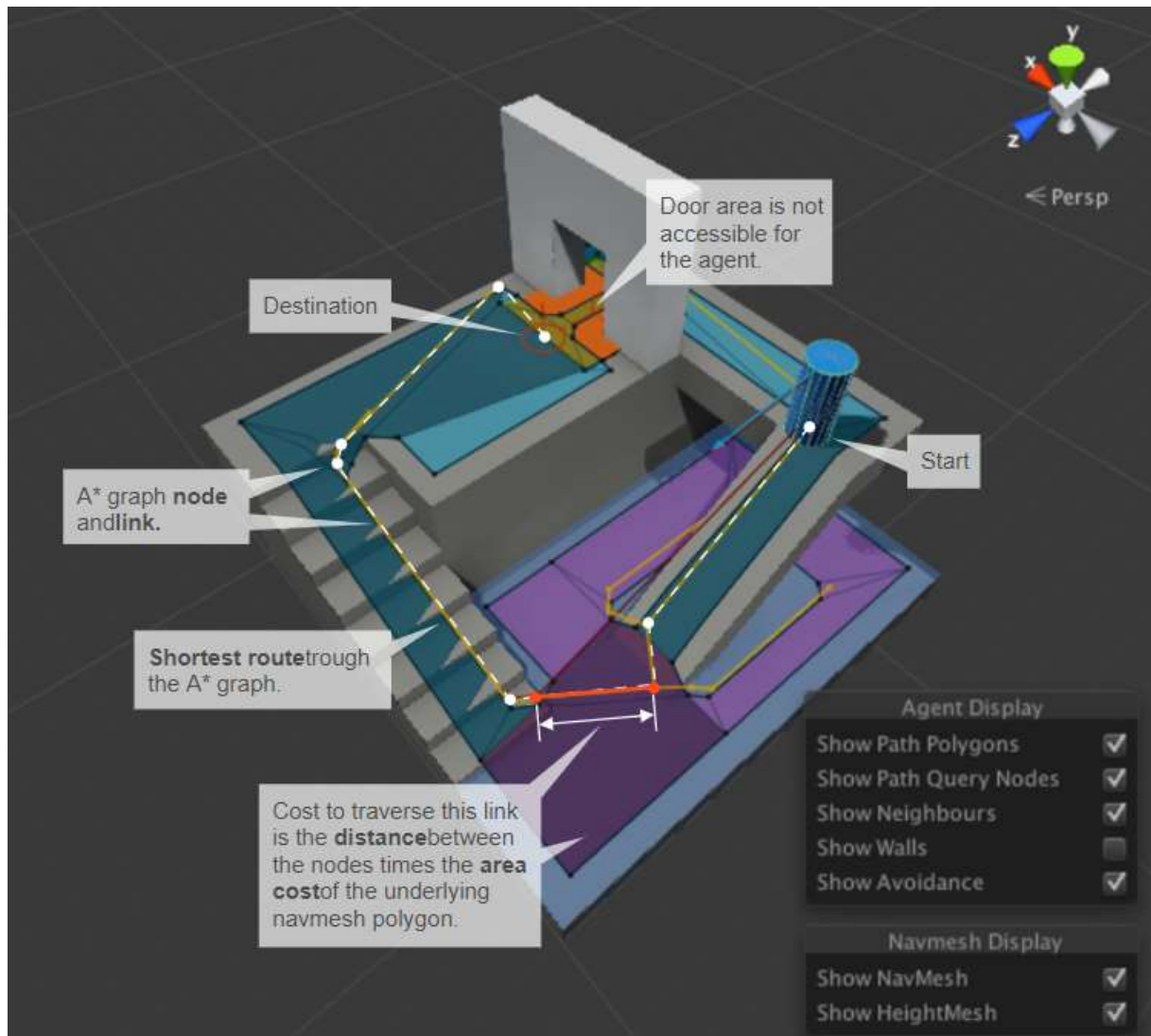


Figura Erro! Fonte de referência não encontrada..23 – Exemplo de um NavMesh complexo aplicado em um cenário com dois andares.
Fonte: Unity (2018).

6.2.3 EXEMPLOS DE IA EM JOGOS ELETRÔNICOS

A inteligência artificial é um campo complexo e amplo, com aplicações em todas as atividades da computação, inclusive nos jogos eletrônicos. Um dos primeiros

jogos eletrônicos, Tennis for Two, foi lançado em 1958, época em que a I.A. era um campo extremamente novo e inexplorado.

O avanço da computação permitiu com que os jogos introduzissem comportamentos de inteligência artificial. Mesmo que simples, isso trazia uma nova dinâmica aos games, pois, além de permitir com que jogadores enfrentassem a famosa CPU, trouxe uma nova camada de desafios e de interação entre jogador e máquina.

6.2.3.1 PAC-MAN

Um dos jogos mais conhecidos do mundo, o Pac-Man, foi um dos primeiros a introduzir uma inteligência artificial relativamente complexa para a época, onde cada um dos personagens controlados pelo computador possuía padrões de movimento diferentes.

O fantasma vermelho, Blinky, persegue o jogador obstinadamente durante o jogo. Já o fantasma rosa, Pinky, se posiciona em um ponto que é 32 pixels a frente da boca do Pac-Man, enquanto o fantasma azul, Inky, tenta se posicionar em um local semelhante. Por fim, Clyde, o fantasma laranja, se movimenta de forma aleatória. Como o Pac-Man está constantemente em movimento, o jogador tem a sensação de ser encurralado pelos fantasmas o tempo inteiro

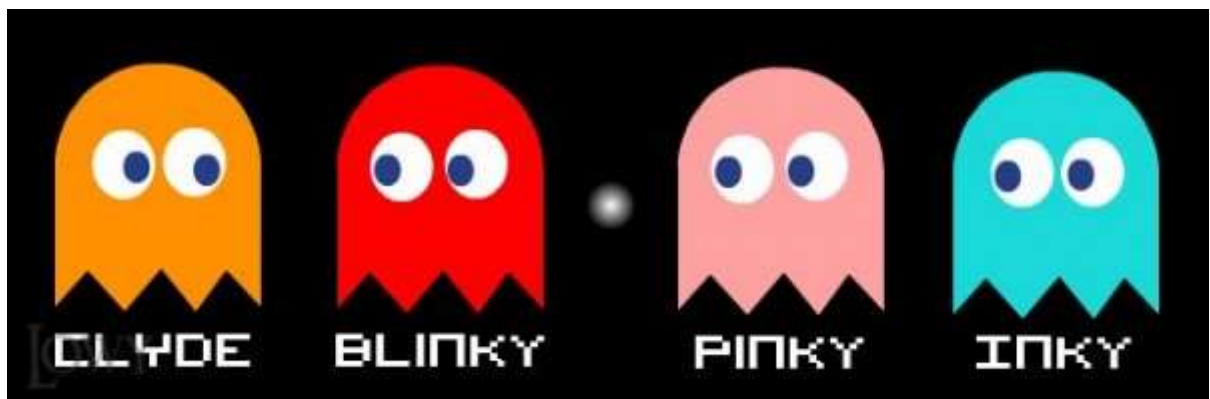


Figura **Erro! Fonte de referência não encontrada.**24 – Personagens de I.A. do jogo Pac-Man.
Fonte: Namco (1980).

6.2.4 IA: DERROTANDO SERES HUMANOS

O avanço da inteligência artificial pôde ser comprovado quando os cientistas resolviam testar a capacidade de processamento do computador e comparar com o

ser humano. O famoso “Teste de Turing” reflete muito bem o anseio de criar um comportamento inteligente para a máquina que se seja equivalente a um ser humano.

6.2.4.1 CUBO MÁGICO

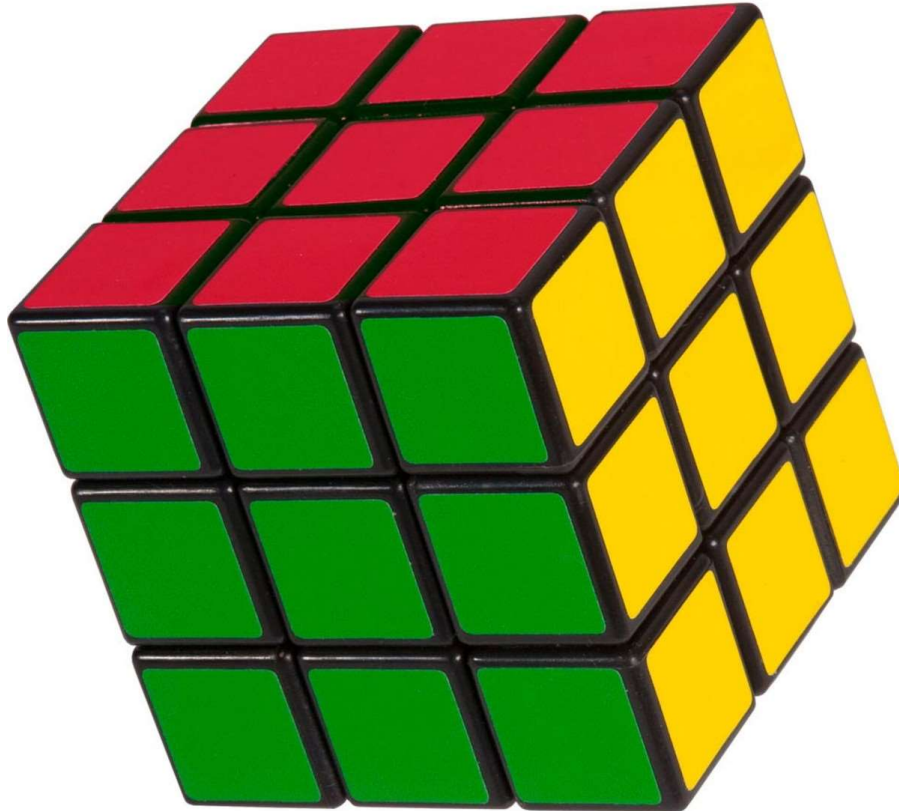


Figura **Erro! Fonte de referência não encontrada.**25 – Cubo Mágico (Rubik's Cube).
Fonte: Rubik's Brand Ltd (1974).

O cubo mágico é um brinquedo famoso pois possui algoritmos complexos de resolução que poucos seres humanos são capazes de masterizar. Para resolver um cubo mágico, um ser humano deve decorar várias combinações de movimentos e aplica-las o mais rápido possível.

Para nós, poderia levar meses para resolver um cubo mágico, porém, com treino, é possível melhorar as habilidades e chegar a velocidades extremamente rápidas. O recorde mundial é do Australiano Feliks Zemdegs, que resolveu a peça em incríveis 4,22 segundos. Entretanto, um robô criado no MIT foi capaz de resolver o mesmo tipo de cubo em 0,38 segundo.



Figura **Erro! Fonte de referência não encontrada..26** – Ser Humano Resolvendo um Cubo Mágico.
Fonte: Feliks Zemdeg (2018).

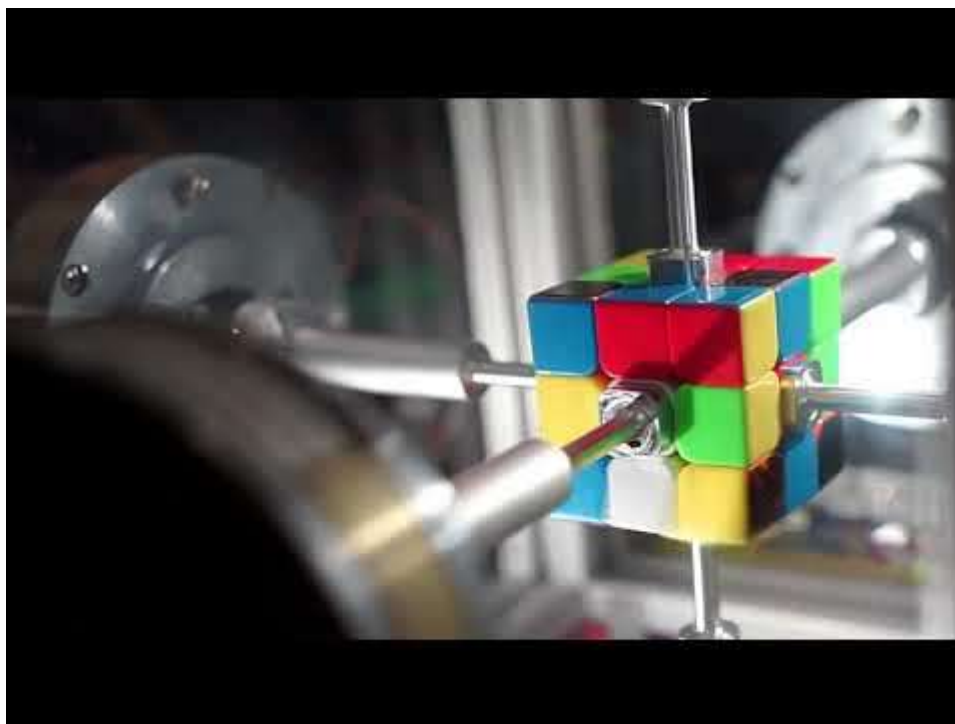


Figura **Erro! Fonte de referência não encontrada..27** – IA Resolvendo um Cubo Mágico.
Fonte: Amazing World (2018).

6.2.4.2 PÔQUER



Figura **Erro! Fonte de referência não encontrada.**28 – Pôquer.
Fonte: Superinteressante (2018).

E quando o assunto é um jogo de cartas mundialmente famoso no qual o blefe é tão relevante quanto ter boas cartas na mão? Uma inteligência artificial desenvolvida em Pittsburg (EUA), derrotou seis adversários chineses durante um torneio de 5 dias, na China. A disputa, em 2017, fez com que a máquina superasse em quase US\$ 800 mil ao final do torneio.

6.2.4.3 XADREZ

O jogo de tabuleiro Xadrez é considerado o primeiro grande desafio para a inteligência artificial no século XX. As primeiras vitórias da máquina sobre o ser humano vêm de 1956, quando o computador MANIAC foi capaz de vencer um novato em 23 movimentos.



Figura **Erro! Fonte de referência não encontrada..29** – Garry Kasparov vs Deep Blue.
Fonte: Scientific American (2017).

Entretanto, apesar desse histórico positivo, as máquinas voltaram a receber atenção em 1996 quando o projeto DeepBlue, da IBM, enfrentou o campeão mundial da modalidade na época. Garry Kasparov e DeepBlue disputaram, e para a surpresa de todos, o computador venceu o primeiro jogo, marcando a primeira vez em que uma máquina havia vencido um campeão mundial. Entretanto, o enxadrista conseguiu encontrar uma falha no algoritmo da máquina e contornou a situação, terminando em 4 a 2 para Garry Kasparov.

Um ano depois, em 1997, ainda mais desenvolvido e preparado, Garry aceitou a revanche e acabou perdendo para a máquina por 3,5 a 2,5. Algumas polêmicas foram criadas na partida, pois em alguns momentos o computador tomou algumas decisões que pareciam estranhas, como sacrificar uma peça para uma compensação a longo prazo, algo que parecia impossível para uma máquina.



Figura **Erro! Fonte de referência não encontrada.**30 – Braço robótico jogando xadrez contra garoto.

Fonte: cs.washington.edu (desconhecido).

6.2.4.4 PERGUNTAS E RESPOSTAS

Um programa muito famoso nos Estados Unidos, o “Jeopardy!” é um show de perguntas e respostas com diversos temas, assim como o Show do Milhão no Brasil, mas em um formato entre múltiplos-jogadores. Em 2011, o supercomputador Watson, criado pela IBM, enfrentou e venceu os então campeões do programa, algo que parecia impossível visto que as perguntas envolviam interpretações difícilíssimas de cultura humana

O programa sorteada perguntas como qual programa de TV antigo teve a aparição de um presidente dos Estados Unidos, algo que a para a máquina responder precisa de uma análise profunda de milhões de páginas da internet, vídeos e até áudios, o que era incompreensível até então.



Figura **Erro! Fonte de referência não encontrada.**31 – Watson contra Ken e Brad em Jeopardy!.
Fonte: The New York Times (2011).

6.2.4.5 GO

Um dos jogos de tabuleiros mais antigos, o Go! foi criado na China mais de mil anos antes de cristo, fazendo com que os humanos tenham mais de 3 mil anos de experiência nesse jogo, que é um dos mais difíceis do mundo.

Ainda assim, em 2016, o projeto AlphaGo, desenvolvido pelo Google, foi capaz de derrotar o sul-coreano campeão mundial Lee Se-dol, por 4 a 1, utilizando estratégias pouco comuns aos jogadores humanos, construindo um jeito completamente inovador de jogar.



Figura **Erro! Fonte de referência não encontrada.**32 – AlphaGo contra Lee Se-dol no jogo chinês Go!.

Fonte: CNN (2016).

6.2.4.6 DOTA

Em 2017 pudemos conhecer o projeto da OpenAI, empresa de Elon Musk, que consistia em uma inteligência artificial capaz de jogar Dota 2, da Valve. A IA, que começou seus treinos perdendo inclusive para amadores, foi capaz de vencer no 1v1 jogadores profissionais.

O projeto continuou em 2018, ainda mais ambicioso, com o objetivo de conseguir vencer profissionais em uma partida 5v5, onde o jogo se torna bem mais complexo. Para que a IA conseguisse vencer, foi selecionado um set de campeões bem mais limitado, 18 dentre os mais de 100 personagens disponíveis, o que limita muito o número de possibilidades que a máquina precisa calcular.



Figura **Erro! Fonte de referência não encontrada..33** – Team Human na partida contra a máquina da OpenAI.

Fonte: OpenAI (2017).

Para que os resultados fossem possíveis, foi utilizado um processamento gigantesco, que possibilitava uma observação de 180 anos por dia para cada campeão, no modo 5v5, totalizando 900 anos por dia de gameplay. Cada campeão possui sua própria rede neural, que não possui um canal específico de comunicação, porém, o trabalho em equipe é coordenado por um parâmetro que vai de 0 a 1 chamado “team spirit”, que define o quanto a IA deve cuidar da recompensa individual ou da equipe em si.

	OPENAI 1V1 BOT	OPENAI FIVE
CPU	60,000 CPU cores on Azure	128,000 preemptible CPU cores on GCP
GPU	256 K80 GPUs on Azure	256 P100 GPUs on GCP
Experience collected	~300 years per day	~180 years per day (~900 years per day counting each hero separately)
Size of observation	~3.3 kB	~36.8 kB
Observations per second of gameplay	10	7.5
Batch size	8,388,608 observations	1,048,576 observations
Batches per minute	~20	~60

Figura **Erro! Fonte de referência não encontrada.**34 – Especificações técnicas do processamento utilizado para os cálculos da IA.
Fonte: OpenAI (2018).

7 REFERÊNCIAS

Unity AR Tutorial: Augmented Reality Game Development with Vuforia, Conteúdo de 6 de Agosto de 2017 <<https://blog.theknightsofunity.com/unity-vuforia-guide/>> Acessado em 02 de Novembro de 2018

Pac-Man: conheça as curiosidades mais interessantes sobre a franquia, Conteúdo de 09 de Novembro de 2017 <<https://www.techtudo.com.br/noticias/noticia/2014/11/pac-man-conheca-curiosidades-mais-interessantes-sobre-franquia.html>> Acessado em 02 de Novembro de 2018

I'll tell you why Deep Learning is so popular and in demand, Conteúdo de 06 de Janeiro de 2018 <<https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780>> Acessado em 02 de Novembro de 2018

History of AI design in video games and its development in RTS games <https://sites.google.com/site/myangelcafe/articles/history_ai> Acessado em 03 de Novembro 2018

Técnicas de Inteligência Artificial em Jogos Eletrônicos, Conteúdo de 20 de Junho de 2013 <<https://pt.slideshare.net/rogeritter/tcnicas-de-inteligencia-artificial-em-jogos-eletrnicos-23242028>> Acessado em 03 de Novembro de 2018

What is the best alternative to A* Algorithm? <<https://www.slant.co/options/11585/alternatives/~a-algorithm-alternatives>> Acessado em 04 de Novembro de 2018

Perdeu, humano! A inteligência artificial já é melhor que você nesses jogos, Conteúdo de 22 de Junho de 2018 <<https://tecnologia.uol.com.br/listas/perdeu-humano-a-inteligencia-artificial-ja-e-melhor-que-voce-nesses-jogos.htm>> Acessado em 04 de Novembro 2018

Inteligência artificial de Elon Musk derrota jogadores experientes de Dota 2, Conteúdo de 2017 <<https://tecnoblog.net/221091/dota-2-inteligencia-artificial-openai/>> Acessado em 04 de Novembro de 2018

Inteligência artificial derrota equipe de jogadores de Dota 2, Conteúdo de 2018 <<https://tecnoblog.net/254596/ia-derrota-jogadores-de-dota-2/>> Acessado em 04 de Novembro de 2018

Blog Open AI – Dota 2, Conteúdo de 11 de Agosto de 2017
<<https://blog.openai.com/dota-2/>> Acessado em 04 de Novembro de 2018

Blog Open AI – OpenAI Five, Conteúdo de 25 de Junho de 2018
<<https://blog.openai.com/openai-five/>> Acessado em 04 de Novembro 2018