






RESEARCH PAPER

Simulação de Processos Cognitivos: Um Estudo de Caso com Animais em Jogos Digitais

Simulation of Cognitive Processes: A Case Study with Animals in Digital Games

Marco Bossle Villanova   [Universidade do Vale do Itajaí - Univali | marco.villanova@edu.univali.br]

Marcelo Dornbusch Lopes   [Universidade do Vale do Itajaí - Univali | marcelo@univali.br]

 Design de Games - Ciência da Computação, Escola Politécnica, Campus Univali Florianópolis, SC, Brasil.

Resumo. Este trabalho apresenta o desenvolvimento de uma inteligência artificial baseada em modelo de controle preditivo, aplicada à simulação de animais em jogos digitais. O principal objetivo é avaliar como essa abordagem pode proporcionar maior autonomia e flexibilidade ao comportamento de personagens virtuais, aproximando o processo de tomada de decisão computacional de características observadas em sistemas biológicos. O modelo proposto é comparado a uma máquina de estados finitos, utilizada como base de referência. A comparação é realizada a partir de métricas coletadas em uma série de simulações controladas, considerando diferentes cenários iniciais e níveis de dificuldade. Os resultados indicam que o modelo baseado em controle preditivo apresenta maior tempo médio de sobrevivência e comportamento mais flexível em comparação à abordagem baseada em estados, especialmente em cenários ideais e intermediários, ao lidar de forma mais adequada com variações do ambiente.

Abstract.

This paper presents the development of an artificial intelligence approach based on Model Predictive Control for the simulation of animals in digital games. The main objective is to assess whether this approach can increase the autonomy and flexibility of virtual character behavior, bringing computational decision making closer to patterns commonly observed in biological systems. The proposed model is compared with a Finite-state Machine baseline. The evaluation is performed through controlled simulations under different initial conditions and difficulty levels. Results indicate that the MPC-based model achieves a longer average survival time and exhibits more flexible behavior than the FSM approach, especially in ideal and intermediate scenarios, responding more effectively to environmental changes.

Palavras-chave: Jogos Digitais, Tomada de Decisão, Inteligência Artificial, Controle Preditivo, Modelos de Comportamento, Simulação de Agentes.

Keywords: Digital Games, Decision Making, Artificial Intelligence, Predictive Control, Behavior Models, Agent Simulation.

Received: 19 December 2025 • **Accepted:** 20 March 2026 • **Published:** 27 March 2026

1 Introdução

Inteligência Artificial (IA) é uma área muito extensa, com aplicações em diversos nichos e que tem crescido mais a cada dia. Seu principal objetivo é dar aos computadores e programas uma capacidade de autonomia, que muda de acordo com a aplicação [Waltham and Moodley, 2016]. Quando utilizada em jogos, ela pode ser usada para dar vida a Personagens Não Jogáveis (*Non-Player Character* — NPC), adicionando uma camada de interatividade e imersão ao jogador [Adeniyi *et al.*, 2024].

A IA em jogos digitais tem evoluído de forma significativa, acompanhando a crescente complexidade em experiências interativas. Comportamentos simples baseados em regras determinísticas são amplamente usados, sendo suficientes para criar um desafio básico ao jogador [Hu *et al.*, 2011]. Porém, à medida que os jogos se tornaram mais imersivos, foi surgindo a necessidade de modelos capazes de lidar com ambientes dinâmicos e de proporcionar comportamentos menos previsíveis, com maior adaptabilidade.

Essa demanda também se manifesta na busca por comportamentos mais coerentes em Personagens Não Jogáveis em cenários realistas, em que decisões inadequadas ao contexto

podem comprometer a experiência do jogador [Sugahara and Lopes, 2021].

No contexto específico da simulação de animais virtuais, a demanda por autonomia e flexibilidade é ainda mais evidente. Diferente de inimigos ou NPCs passivos tradicionais, animais em ambientes digitais devem apresentar comportamentos variados, adaptativos e compatíveis com a expectativa do jogador em relação à vida real [Naik *et al.*, 2020]. Para isso, diferentes abordagens têm sido empregadas, cada uma com suas vantagens e limitações.

Além disso, abordagens de simulação baseadas em agentes e técnicas de vida artificial em jogos têm sido exploradas para ampliar a dinâmica e a responsividade de entidades autônomas em ambientes interativos, como apresentado em [Martins *et al.*, 2023].

Entre as abordagens mais consolidadas, está a Máquina de Estados Finitos (*Finite State Machine* — FSM), utilizada em jogos clássicos como PAC-MAN [Jagdale, 2021] por diversos fatores positivos, como simplicidade de implementação, boa performance em tempo real e previsibilidade [Nugraha *et al.*, 2025]. Porém, esse sistema tende a apresentar comportamentos rígidos e pouco adaptativos, com pouca escalabilidade à medida que aumenta a complexidade das condições e

transições. Como alternativa, uma técnica de controle mais sofisticada, como *Model Predictive Control* (MPC), se mostra promissora no objetivo de criar interações mais inteligentes [Sani *et al.*, 2021], assim, aumentando a verossimilhança com comportamentos reais.

Da mesma forma que o conceito de *Simple Predictive Control* (SPC) por Wawrzyński *et al.* [2008], o MPC se destaca por sua capacidade de considerar o estado atual, prever consequências futuras das ações e selecionar alternativas que maximizam a eficiência e desempenho do agente. Essas previsões — que acontecem dentro do horizonte de predição [Negenborn *et al.*, 2004] — são o resultado de um cálculo que aplica um custo e ganho para cada ação, assim escolhendo continuamente a alternativa mais eficiente. O conjunto dessas características aproxima o processo de tomada de decisão artificial de padrões discutidos na literatura sobre sistemas biológicos [Saberri Moghadam *et al.*, 2019], permitindo um comportamento mais flexível e adaptativo.

Dito isso, este trabalho tem como objetivo investigar a aplicação do MPC no desenvolvimento de uma IA para animais em jogos digitais. A proposta é comparar uma FSM, que servirá de referência, a partir de métricas coletadas em simulações controladas. O resultado esperado é que a análise comparativa permita observar em que condições o MPC proporciona maior autonomia e flexibilidade ao comportamento do agente, em comparação à abordagem baseada em FSM.

2 Caracterização da pesquisa

Pesquisas em Ciência da Computação podem ser caracterizadas segundo sua natureza e estratégia de validação. Conforme Wazlawick [2009], uma parte significativa da produção na área é empírica, avaliando propostas por meio de experimentação e evidências quantitativas, especialmente quando o objeto de estudo envolve sistemas executáveis e comparação de desempenho. Complementarmente, Gil [2017] propõe classificar pesquisas quanto à natureza (básica/aplicada), aos objetivos (exploratória, descritiva, explicativa) e aos procedimentos técnicos (por exemplo, experimental).

Quanto à natureza, este trabalho é aplicado, pois investiga a utilização de um mecanismo de tomada de decisão em um problema específico do desenvolvimento de jogos digitais: a simulação de comportamento de agentes do tipo animal em ambiente 2D. Quanto aos objetivos, trata-se de um estudo predominantemente explicativo no sentido de investigar o efeito do mecanismo de decisão sobre métricas observáveis de desempenho e comportamento; ao mesmo tempo, mantém um caráter exploratório por avaliar condições de contorno e sensibilidade a cenários iniciais. A abordagem é quantitativa, baseada em medidas coletadas ao longo de execuções repetidas.

O delineamento adotado é experimental e comparativo, com simulações controladas e repetição de execuções independentes. O experimento pode ser descrito como um arranjo fatorial 2×3 , no qual se avaliam dois modelos de decisão (FSM e MPC) sob três condições iniciais (cenários C1, C2 e C3), com múltiplas repetições por combinação. O objetivo do delineamento é isolar o efeito dos fatores sobre as métricas observadas, mantendo constantes as demais características do ambiente.

As variáveis independentes (fatores) são: (i) o modelo de tomada de decisão, com dois níveis (FSM e MPC); e (ii) o cenário, com três níveis (C1, C2, C3), que diferem pelas condições iniciais dos status do agente. As variáveis dependentes são as métricas registradas ao final de cada execução, incluindo tempo de sobrevivência e estatísticas dos status (por exemplo, vida, fome e sono), além de medidas de atividade (por exemplo, quantidade de ações ou ações por segundo). Como variáveis de controle, mantêm-se fixos o ambiente e suas regras, o conjunto de ações disponíveis, as condições de término da simulação e, no caso do MPC, sua parametrização (por exemplo, horizonte de predição, fator de desconto e pesos), de modo a garantir uma comparação justa entre as abordagens.

Por fim, ressalta-se que os resultados refletem o ambiente e a parametrização avaliados neste estudo de caso; a generalização para outros jogos e ou ambientes depende de validações adicionais.

3 Trabalhos relacionados

Diversos estudos têm explorado a aplicação de modelos de controle e tomada de decisão em sistemas, predominantemente em contextos industriais e robóticos, e em menor medida em jogos digitais. Apesar dessa diferença, os principais conceitos de MPC podem ser aplicados na implementação de agentes para jogos. A teoria deste trabalho se apoia em abordagens que utilizam MPC como alternativa às arquiteturas tradicionais, como FSMs. Por isso, esta seção tem como objetivo analisar trabalhos com objetivos e conceitos similares.

O principal trabalho usado como referência é o de Wawrzyński *et al.* [2008], que aplica o conceito de controle preditivo à inteligência artificial em jogos. Os autores propõem um agente — (implementado no jogo Half-Life) capaz de gerar e avaliar planos de ação em tempo real com base em retornos previstos, utilizando uma arquitetura hierárquica dividida em níveis inferior e superior. O nível inferior é responsável por executar o comportamento do agente, como ir de um ponto a outro. Já o nível superior toma as decisões, planejando o futuro e selecionando a próxima tarefa, atribuindo um custo para cada possível tarefa dentro do horizonte de predição e maximizando o lucro. O modelo resultante foi comparado com LedgeWalker, o NPC nativo do jogo, e apresentou um comportamento mais adaptativo, eficiente e "human like", de acordo com o autor.

Ainda sobre controle preditivo, porém fora da área de jogos, Negenborn *et al.* [2004] realizaram uma revisão sobre o MPC aplicado em sistemas multiagente, explorando diferentes formas de coordenação entre agentes. O estudo destaca que arquiteturas distribuídas (com comunicação entre os agentes) e hierárquicas (com custos dependendo do nível hierárquico) podem reduzir o custo computacional e aumentar a escalabilidade, mantendo a estabilidade global. Esses princípios podem influenciar o uso de estruturas semelhantes em jogos, onde múltiplos agentes interagem em um mesmo ambiente e precisam alinhar objetivos locais e globais. Um exemplo prático dessa aplicação é uma horda de inimigos ou um rebanho de animais, onde todos os agentes possuem um objetivo individual e grupal.

Complementando a ideia de MPC multiagente, Maes-

tre *et al.* [2011] propuseram uma variação cooperativa do MPC distribuído, baseada na teoria de jogos cooperativos, para coordenar múltiplos agentes controladores. Esses agentes, divididos em subsistemas, dividem recursos e restrições, cooperando para aproximar o desempenho de um MPC centralizado, mas com custo computacional reduzido e maior escalabilidade. Essa perspectiva enfatiza o equilíbrio entre agentes e a importância de comunicação eficiente, uma ideia que pode ser estendida ao mimetismo de comportamentos sociais ou coletivos em simulações com múltiplos personagens. Em conjunto, esses trabalhos demonstram o controle preditivo como um conceito capaz de unir eficiência, adaptabilidade e autonomia, utilizando conceitos como horizonte de predição e cálculo de custo. Contudo, também apontam limitações relevantes, como custo computacional e dependência de modelos/heurísticas adequados ao domínio, o que pode restringir sua aplicação em sistemas com recursos limitados. Uma comparação direta dos trabalhos com a proposta atual é apresentada na **Tabela 1**.

Tabela 1. Trabalhos relacionados e relação com a proposta deste artigo.

Trabalho	Contexto	Relação com este trabalho
Wawrzyński <i>et al.</i> [2008]	IA para jogos (Half-Life)	Motiva o uso de controle preditivo em jogos e a definição de horizonte e custo como alternativa à FSM.
Negenborn <i>et al.</i> [2004]	Revisão: MPC multiagente	Sustenta a discussão de custo e escalabilidade e aponta extensões para múltiplos agentes (grupos de animais).
Maestre <i>et al.</i> [2011]	MPC distribuído cooperativo	Inspira extensões cooperativas/coletivas, embora o escopo atual seja single-agent.
Este trabalho	Simulação de animais em jogo 2D	Compara controle preditivo e FSM em simulação 2D com métricas em cenários controlados.

Apesar das evidências apontadas na literatura, ainda há relativamente poucos estudos que avaliem, de forma quantitativa e controlada, o uso de controle preditivo como mecanismo de tomada de decisão em simulações de animais em jogos digitais, comparando-o diretamente com abordagens tradicionais baseadas em estados. Nesse sentido, este trabalho contribui com um estudo de caso implementado em ambiente de jogo 2D, no qual controle preditivo e FSM são comparados a partir de métricas coletadas em múltiplos cenários de simulação.

4 Definição do modelo

O MPC pode ser considerado uma técnica [Sani *et al.*, 2021] de controle que utiliza previsões de estados futuros para determinar a ação mais eficiente. A partir da observação do estado atual, ele projeta possíveis evoluções futuras por meio de um modelo de transição e seleciona a ação (ou sequência de ações) que minimiza um custo ou maximiza um ganho

[Negenborn *et al.*, 2004]. O número máximo de iterações dessa projeção é definido pelo horizonte de predição. Cada ação (ou estado, dependendo da implementação) é testada em um futuro dentro do horizonte de predição, aplicando as variações impostas pelo sistema nas variáveis avaliadas e, ao final, são somados os custos (ou ganhos) de cada predição [Wawrzyński *et al.*, 2008].

Neste trabalho, o MPC é implementado de forma discreta e baseada em simulação: para cada ação possível, o agente cria uma cópia temporária do estado atual e simula sua progressão ao longo do horizonte de predição, aplicando heurísticas de degradação, recuperação de status e/ou eventos do ambiente. Ao final, a ação selecionada é aquela que apresenta o menor custo acumulado.

De forma resumida, para cada ação candidata $a \in A$, calcula-se:

$$J(a) = \sum_{t=0}^{H-1} \gamma^t \text{StepCost}(s_t, a), \quad (1)$$

onde H é o horizonte de predição e γ é o fator de desconto. Ao final, a ação escolhida é $a^* = \arg \min_{a \in A} J(a)$.

Por exemplo, imagine um NPC com barras de vida e fome, além de um peso atribuído a cada uma dessas variáveis. Seu objetivo é comer, fugir de predadores ou permanecer inativo. Empregando a lógica do MPC, o agente analisa todas as ações possíveis (neste caso, três) e simula as consequências de cada uma delas, dentro do horizonte de predição. Durante essa simulação, são aplicadas as variações do sistema — como diminuição gradual da fome, presença de inimigos (que resulta em combate) e presença de alimentos — e calculado o custo de cada cenário, com base nos status modificados por seus respectivos pesos. Após todas as predições, a ação com o menor custo é selecionada, por ser a mais favorável. Adicionalmente, pode-se aplicar um fator de desconto (**gamma**) na soma dos custos durante as iterações, balanceando a importância entre resultados imediatos e consequências futuras. Para auxiliar na visualização, foi montada a **Figura 1**, que representa o fluxo da lógica descrita.

Já a FSM é uma arquitetura de código comumente usada na indústria de jogos, principalmente na implementação de NPCs. Ela divide os comportamentos de um agente em estados controlados por condições que, quando atendidas, acionam um próximo passo [Jagdale, 2021]. Essa abordagem possui diversas vantagens e desvantagens, sendo as principais vantagens o baixo custo computacional, evidenciado pelo uso em jogos antigos, o bom desempenho em sistemas mais simples e a facilidade de implementação [Nugraha *et al.*, 2025]. Por outro lado, quanto maior a complexidade do sistema, mais estados e ações precisam ser implementadas para que a IA se adapte ao ambiente, algo que os autores Hu *et al.* [2011] chamam de "combinatorial explosions".

Comparativamente, o MPC apresenta um nível de flexibilidade maior do que a FSM, pois não depende de transições fixas. Enquanto a FSM reage a condições predefinidas de forma determinística, o MPC avalia múltiplas possibilidades futuras, escolhendo a ação mais adequada. Porém, o MPC requer maior capacidade de processamento, ficando mais complexo de implementar em tempo real. Por isso, pode-se considerar que a FSM prioriza simplicidade e eficiência, enquanto

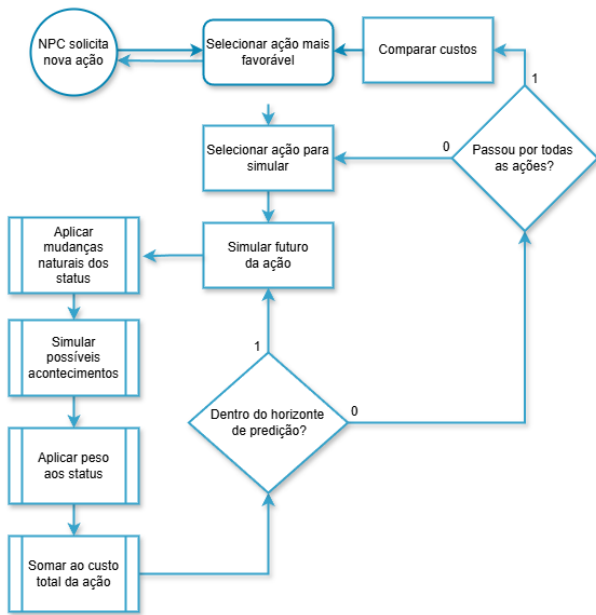


Figura 1. Fluxograma do MPC.

```

1  override protected AIAction DecideNewAction()
2  {
3      AIAction newAction = AIAction.NONE;
4
5      switch (currentAction)
6      {
7          case AIAction.NONE:
8              newAction = AIAction.SEARCHINGFOOD;
9              break;
10         case AIAction.SEARCHINGFOOD:
11             newAction = nearestFoodSource ?
12                 AIAction.EATING :
13                 AIAction.SEARCHINGFOOD;
14             break;
15         case AIAction.EATING:
16             if (ai_status.sleep < ai_status.hunger)
17                 newAction = AIAction.SLEEPING;
18             else
19                 newAction = AIAction.SEARCHINGFOOD;
20             break;
21             // ...
22         }
23     }
24     return newAction;
25 }
    
```

Código 1: Trecho da lógica de transição de estados da FSM.

o MPC busca autonomia e flexibilidade no comportamento do agente.

4.1 Código

Para a implementação das IAs, foi utilizada a Unity Engine, especificamente a versão 6.0. Foram definidos três critérios para escolha de motor: 1. Uso gratuito; 2. Ambiente 2D nativo; 3. Documentação extensa. Foram identificadas duas engines que se encaixavam nesses critérios, elas sendo a Unity e Godot. Por conta de ambas se encaixarem nos critérios, foi escolhida a Unity.

Como definido anteriormente, a principal diferença entre as IAs será a tomada de decisões. Portanto, como o comportamento dos estados serão iguais, a estrutura foi dividida em uma classe principal denominada AIHead, com outras duas que herdam da mesma, chamadas FSM e MPC.

A classe principal contém toda a lógica comum aos agentes, como variáveis usadas no controle, a execução da ação atual e os métodos virtuais para as decisões. O fluxo dos comportamentos, como andar em uma direção, acaba sendo irrelevante para a tomada de decisões, levando em conta que estas são baseadas nos status finais de cada iteração do horizonte. Portanto, esse fluxo de comportamento é escondido da FSM e MPC, sendo gerenciado a cada quadro na classe AIHead, dentro do método privado Update, uma função nativa da Unity que é chamada antes da etapa de renderização [Unity Technologies, 2025].

A FSM, assim como definido por Jagdale [2021] e Nagraha et al. [2025], utiliza transições condicionais. A cada término de ação, a classe determina qual será a próxima ação, utilizando condições contextualizadas. Esse modelo é eficiente (computacionalmente falando) e simples, mas pode se tornar rígido dependendo da complexidade condicional. Essa lógica, aplicada ao contexto do agente, foi utilizada na construção do código, com um trecho podendo ser observado no Código 1.

Essa lógica é direta, mas não considera previsões de longo prazo, apenas reage ao que percebe no momento da

transição.

A classe MPC, por outro lado, substitui as transições determinísticas por um processo baseado em simulação, onde cada ação possível é avaliada antes de ser executada. Para isso, o modelo cria uma cópia temporária do estado atual do agente, utilizando uma estrutura denominada de SimulatedStatus, que contém os valores de saúde, sono, fome e o máximo permitido por status. Essa cópia é inicializada utilizando como parâmetro uma referência do estado real do agente e, em seguida, passa por uma série de iterações ao longo do horizonte de predição, nas quais são aplicadas as heurísticas de progressão.

Em cada iteração, é calculado um custo utilizando a função StepCost. Esse custo é composto pela soma de quatro valores relacionados a um aspecto do agente. Para fome, sono e saúde, o cálculo segue o mesmo princípio: é comparado o valor atual desse atributo com seu valor ideal (max), é normalizada essa diferença dividindo por max e, em seguida, o valor é elevado ao quadrado para ampliar penalizações e multiplicado por seus respectivos pesos (wHunger, wSleep, wHealth).

A única diferença é no caso do sono, que é aplicado um custo extremamente alto quando o nível do sono estiver cheio, forçando o MPC a focar em outras ações. Por fim, o wEnemy adiciona custo associado à condição de ameaça considerada na simulação (por exemplo, presença de inimigos), contribuindo para priorizar ações de resposta. A lógica descrita pode ser visualizada no Código 2, que contém o código utilizado para cálculo de custo.

Após cada etapa simulada é gerado um custo, que representa o quão indesejável aquele estado futuro é. Esses custos são acumulados e ponderados por um fator de desconto (gamma), que controla o peso relativo entre efeitos imediatos e consequências futuras. Ao final, a ação que apresenta o menor custo total é selecionada como próxima. Para facilitar o entendimento do fluxo de comportamento da IA, foi montada a Figura 2.

Em resumo, a FSM prioriza eficiência e resultados imediatos, enquanto o MPC prioriza autonomia e flexibilidade.

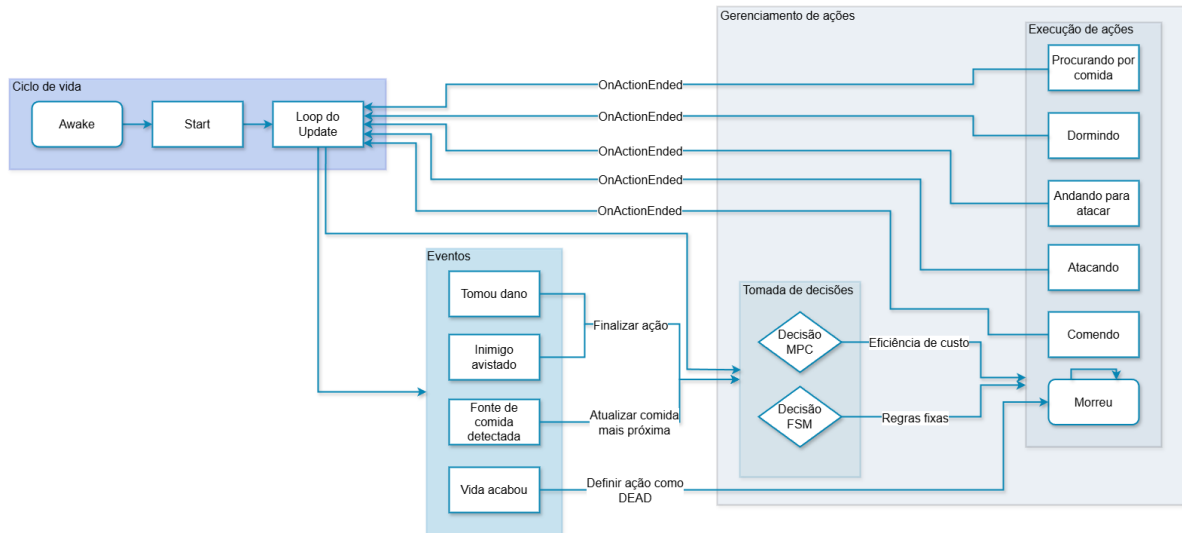


Figura 2. Fluxo de comportamento da IA

```

1 private float StepCost(ref SimulatedStatus sim,
2   AIAction testAction)
3 {
4   float c = 0.0f;
5   float max = sim.max;
6
7   c += wHunger * Mathf.Pow((max - sim.hunger) /
8     max, 2);
9   c += wSleep * (sim.sleep >= (max - 1.0f) ?
10     99999.0f : Mathf.Pow((max - sim.sleep) / max, 2));
11   c += wHealth * Mathf.Pow((max - sim.health) /
12     max, 2);
13   c += wEnemy * (nearestEnemy == null ? 0.0f : 1.0f);
14   return c;
15 }
    
```

Código 2: Cálculo do custo por etapa (StepCost) no MPC.

A FSM é adequada para jogos em que previsibilidade e baixo custo computacional são prioridades, já o MPC se destaca em simulações que exigem adaptabilidade a ambientes dinâmicos.

5 Design de experimentos

Para avaliar o desempenho das duas abordagens de tomada de decisão, foram realizadas simulações em um ambiente fechado e controlado. Cada simulação representa um ciclo de vida do agente, onde o mesmo executa ações como se alimentar, dormir e reagir a ameaças, até que sua saúde restante chegue a zero.

Com o objetivo de comparar o modelo em diferentes situações, foram definidos três cenários de início para as simulações, onde foram realizadas 100 iterações independentes por modelo, totalizando 600 execuções. Durante as simulações, foram registrados: 1. o tempo total de sobrevivência do agente; 2. a média de valores dos status por simulação; 3. a quantidade de vezes que houve uma troca de ação; 4. o maior e menor tempo de sobrevivência observados. Esses dados serão então somados entre todas as execuções de cada cenário, gerando uma média que será usada em uma análise quantitativa, comparando ao desempenho das escolhas do agente.

Tabela 2. Parâmetros do controle preditivo utilizados nos experimentos.

Parâmetro	Valor
Horizonte (H)	(5.0)
Desconto (γ)	(0.75)
w_{Hunger}	(12)
w_{Sleep}	(0.0001)
w_{Health}	(18)
w_{Enemy}	(14)

Os parâmetros do modelo baseado em controle preditivo foram mantidos fixos ao longo de todos os cenários, de modo a isolar o impacto das condições iniciais dos status sobre o comportamento do agente. A Tabela 2 apresenta os valores utilizados.

Os cenários variaram apenas nos valores iniciais dos status do agente, a fim de representar três níveis de dificuldade, divididos em um estado ideal, um estado neutro e um estado crítico. Ao manter o ambiente e as regras iguais para ambas as IAs, a estratégia de tomada de decisões pode ser melhor observada.

5.1 Cenário 1

O primeiro experimento tem como objetivo avaliar o comportamento das duas abordagens em um ambiente ideal, onde o agente inicia a simulação com todos os status cheios. Este cenário tem como propósito analisar a estabilidade e eficiência das decisões quando não há fatores críticos influenciando o comportamento, permitindo a observação de um ritmo natural de degradação e capacidade de manutenção do equilíbrio por cada modelo.

Após a execução dos 200 ciclos do primeiro cenário, os dados obtidos foram exportados, representados na Tabela 3, com valores arredondados para a segunda casa decimal.

Durante as simulações, a FSM mostrou um comportamento estável, alternando entre comer e dormir de forma cíclica. Já o MPC exibiu transições mais frequentes, alternando entre dormir e comer com maior frequência. A FSM obteve um tempo médio de sobrevivência de 193,20 segundos, enquanto o MPC alcançou 291,41 segundos, um aumento de aproximadamente 51%. Apesar da média dos status ser

Tabela 3. Resultados comparativos do primeiro cenário.

Métrica	FSM	MPC
Tempo médio	193,20	291,41
Vida média	79,37	77,92
Fome média	75,77	75,22
Sono médio	78,17	54,18
Média de ações	63,06	107,09
Menor tempo	32,65	132,75
Maior tempo	405,04	427,71

Tabela 4. Resultados comparativos do segundo cenário.

Métrica	FSM	MPC
Tempo médio	122,71	135,45
Vida média	68,42	74,80
Fome média	58,52	57,38
Sono médio	27,16	27,73
Média de ações	44,97	49,95
Menor tempo	30,91	31,25
Maior tempo	231,17	229,09

menor, o MPC apresentou um maior número de ações (resultado possivelmente influenciado pelo tempo de sobrevivência maior) e uma sobrevivência mais estável, com o menor tempo registrado sendo quatro vezes maior.

Em um ambiente sem adversidades, o MPC confirmou a expectativa teórica de oferecer maior adaptabilidade e autonomia, enquanto a FSM se manteve eficiente, mas restrita às condições determinísticas.

5.2 Cenário 2

Neste cenário, o agente iniciou com todos os status pela metade, simulando uma situação de dificuldade intermediária. O objetivo foi observar como cada IA reagiria à necessidade imediata de restabelecer os status e prolongar a sobrevivência. Com os dados obtidos das 200 simulações, foi montada a **Tabela 4**.

Durante a simulação, tanto a FSM quanto o MPC apresentaram desempenhos semelhantes, com o MPC mantendo uma leve vantagem no tempo médio de sobrevivência e um valor máximo ligeiramente menor. Ao contrário do primeiro cenário, a FSM apresentou uma vida e sono médios — apesar de ser pouco — mais baixos, indicando uma recuperação mais eficiente pelo MPC. Apesar disso, o número de ações foi discretamente maior, sugerindo que o ganho de desempenho veio da ordem e do momento em que as decisões foram tomadas, e não da quantidade de trocas de estado.

5.3 Cenário 3

No terceiro cenário, os agentes iniciaram com fome e sono zerados, representando uma situação crítica desde o início. O propósito foi avaliar a capacidade de resposta das IAs sob condições de emergência, onde há pouca chance de sobrevivência. As métricas registradas do terceiro e último cenário podem ser visualizadas na **Tabela 5**.

Os resultados mostram que ambos os modelos obtiveram baixo tempo médio de sobrevivência, com um desempenho próximo e uma clara vantagem do MPC nos menores e maiores tempos registrados. O MPC novamente manteve uma leve vantagem, resistindo por mais tempo e apresentando maior número de ações, o que pode ser um reflexo de tentativas mais dinâmicas de recuperação, levando em conta os tempos

Tabela 5. Resultados comparativos do terceiro cenário.

Métrica	FSM	MPC
Tempo médio	31,89	33,92
Vida média	74,40	71,90
Fome média	13,16	13,26
Sono médio	0,00	0,03
Média de ações	6,29	8,96
Menor tempo	23,61	27,97
Maior tempo	48,29	63,13

aproximados.

Ao contrário dos outros dois cenários, a FSM atingiu uma média de fome menor. Com isso, nenhum dos modelos conseguiu estabilizar os status de forma eficiente, evidenciado pela disparidade dos status. Esse comportamento sugere que, no regime crítico, a política baseada em estados pode ser sensível à parametrização das condições e prioridades de transição, o que abre espaço para testes adicionais e refinamento da abordagem de referência.

5.4 Discussão dos resultados

O **Gráfico 1** apresenta uma visão consolidada do tempo de sobrevivência nos três cenários avaliados, comparando as duas abordagens. Observa-se que o ganho do controle preditivo é mais pronunciado no primeiro cenário (condições iniciais ideais), no qual há maior margem para o planejamento antecipar degradações e evitar estados indesejáveis. No segundo cenário (intermediário), a vantagem permanece, porém de forma mais modesta, sugerindo que ambos os modelos passam mais tempo em um regime de recuperação reativa, reduzindo o impacto de decisões de longo prazo.

No terceiro cenário (crítico), os resultados indicam que ambas as abordagens têm desempenho limitado, com tempos médios de sobrevivência próximos. Nesse regime, a severidade das condições iniciais tende a dominar a dinâmica do sistema, e pequenas diferenças na tomada de decisão têm menor efeito sobre o desfecho. Ainda assim, a visualização também ajuda a destacar a variabilidade entre execuções: a amplitude do intervalo mínimo–máximo evidencia que alguns episódios evoluem de maneira substancialmente melhor do que outros, mesmo sob o mesmo cenário.

Complementarmente, o **Gráfico 2** resume a média de ações executadas por execução. Em geral, o modelo baseado em controle preditivo apresenta maior atividade ao longo da simulação, o que é consistente com um processo de decisão que avalia alternativas a cada ciclo e pode replanejar com maior frequência. Por outro lado, a FSM tende a executar menos ações, refletindo uma política mais direta e reativa. Em conjunto, essas evidências reforçam o trade-off entre flexibilidade comportamental e custo/complexidade de decisão, particularmente relevante em ambientes de jogo em tempo real.

A análise dos três cenários evidencia algumas diferenças entre as abordagens de decisão, tanto em desempenho quanto em comportamento. De modo geral, o MPC apresentou vantagens em todos os casos, com intensidades distintas conforme o nível de dificuldade do ambiente.

No primeiro cenário, em condições ideais, o MPC demonstrou um tempo médio de sobrevivência melhor ($\approx 51\%$ maior) e um comportamento mais ativo, com um aumento de aproximadamente 69% nas ações. Isso indica uma IA mais

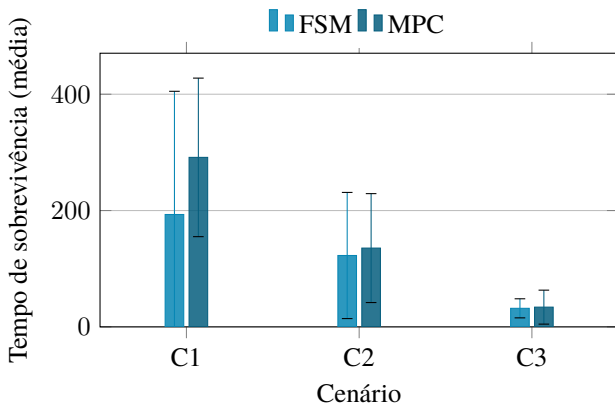


Gráfico 1: Tempo médio de sobrevivência por cenário. As barras de erro representam o intervalo mínimo–máximo observado nas execuções.

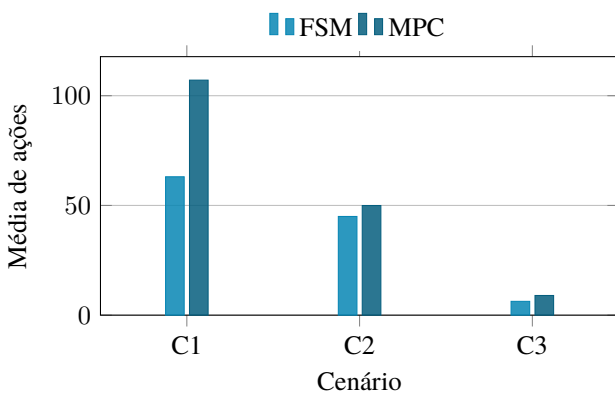


Gráfico 2: Média de ações por cenário (proxy de atividade/decisão ao longo da simulação).

proativa e adaptativa, capaz de antecipar variações nos status e reagir preventivamente. Já a FSM manteve um comportamento mais estável, focando em manter todos os status no mesmo nível, o que resultou em menor autonomia e tempo de vida.

Já no segundo cenário, agora com condições iniciais de dificuldade mediana, as diferenças entre os modelos diminuíram. Ambos os agentes alcançaram um desempenho semelhante, com o MPC ainda apresentando uma vantagem em vida e tempo médio de sobrevivência.

Neste cenário intermediário, o ganho do MPC foi mais sutil do que no cenário ideal: o tempo médio de sobrevivência aumentou em torno de 10%, e a vantagem em vida média foi de aproximadamente 9%. Isso sugere que, quando os status iniciais já começam degradados, ambos os modelos passam mais tempo em um regime de recuperação reativa, reduzindo o espaço para ganhos expressivos de planejamento. Ainda assim, a maior vida média do MPC indica decisões ligeiramente mais eficientes na priorização das ações que evitam quedas de saúde. O fato de a FSM apresentar um valor máximo marginalmente superior, apesar de média menor, sugere maior variabilidade entre execuções, enquanto o MPC tende a resultados mais consistentes.

No terceiro cenário, com condições iniciais críticas, o impacto do modelo de decisão foi limitado. Ambos os agentes tiveram dificuldade em restaurar os status de forma eficaz.

Ainda assim, o MPC manteve ligeira vantagem no tempo total e na variação de ações, mais uma indicação de um comportamento mais dinâmico, mesmo sem sucesso na recuperação. Essa proximidade de resultados aponta que, em situações extremas, as condições iniciais possuem peso determinante sobre o desempenho, reduzindo a influência do algoritmo de decisão.

Um ponto relevante no cenário crítico é que a FSM não acionou o estado de sono em nenhuma execução, refletido no sono médio igual a 0,00. Esse resultado sugere que a lógica condicional pode ter priorizado outras ações e que o agente não chegou a satisfazer as condições para dormir antes do término das execuções, o que reduz a capacidade de recuperação do status de sono nesse regime.

De forma geral, os experimentos mostram que o MPC proporciona maior autonomia, flexibilidade e estabilidade em contextos controlados ou levemente desafiadores, enquanto a FSM se destaca pela simplicidade e eficiência computacional. Porém, esse desempenho superior do MPC vem acompanhado de maior custo computacional, devido às diversas iterações e operações matemáticas envolvidas a cada quadro. Por outro lado, o comportamento da IA pode ser facilmente modificado, somente alterando os valores de peso, horizonte de previsão e gamma. Com esses dados, pode-se concluir que o MPC é mais adequado para ambientes controlados, que buscam flexibilidade e adaptabilidade, enquanto a FSM permanece eficiente em cenários previsíveis e de baixa complexidade.

6 Conclusão

Este trabalho investigou a aplicação do Modelo de Controle Preditivo (MPC) na tomada de decisões de animais em jogos digitais, comparando diretamente com o desempenho de uma Máquina de Estados Finitos (FSM). O objetivo foi avaliar se o MPC poderia oferecer maior autonomia e flexibilidade, aproximando a tomada de decisão de padrões observados em sistemas biológicos, sem implicar equivalência direta.

Ao longo do estudo, foram retomados os conceitos fundamentais das duas abordagens, para detalhar a implementação e definir os cenários de experimentação. Esses cenários permitiram observar como cada IA reage a ambientes com dificuldades variadas, mantendo o ambiente e as regras iguais entre as abordagens, de modo a possibilitar uma comparação justa.

Os resultados evidenciam que o MPC apresentou desempenho superior nos três cenários, com maior destaque no ambiente ideal, onde o tempo médio de sobrevivência foi $\approx 51\%$ maior. Em termos de execução, a FSM apresentou maior taxa média de ações por segundo (3,07) em comparação ao MPC (2,70), o que é consistente com uma abordagem mais simples e eficiente computacionalmente. Em condições moderadas, a vantagem do MPC permaneceu de forma discreta, enquanto no cenário crítico ambos os modelos apresentaram dificuldade em recuperar os status iniciais. Ainda assim, o MPC manteve leve vantagem, apresentando comportamento mais dinâmico e melhor distribuição de ações ao longo do tempo.

Esses dados reforçam a ideia de que o MPC tende a apresentar comportamento mais flexível e autônomo, enquanto a FSM, mesmo eficiente em termos de performance e uso

de recursos da máquina, pode se tornar restrita por depender de transições determinísticas. Como consequência, maior adaptabilidade nas decisões — como discutido por Naik *et al.* [2020] — pode contribuir para a percepção de realismo por parte do jogador.

Todavia, algumas limitações devem ser consideradas. O ambiente utilizado é simples e altamente controlado, o que reduz variáveis externas que poderiam influenciar o comportamento do agente. Além disso, diferentes combinações de pesos podem alterar significativamente os resultados. No cenário crítico, também se observou que a FSM foi sensível às condições e prioridades de transição, o que pode restringir a recuperação de determinados status nesse regime.

Como propostas de continuidade, pode-se comparar o MPC com outros modelos de decisão — como Lógica Fuzzy — e testar as abordagens em ambientes mais complexos, com elementos dinâmicos e situações que exijam coordenação, incluindo a extensão da simulação para 3D. Outra possibilidade relevante é explorar o uso de MPC em grupos de animais (por exemplo, comportamentos de *flocking*), alinhando-se a discussões sobre gerenciamento multiagente presentes na literatura [Negenborn *et al.*, 2004]. Além disso, é pertinente investigar métodos de ajuste automático dos parâmetros do MPC, buscando reduzir dependência de valores definidos manualmente. Bem como a execução de análises estatísticas mais extensas, incluindo desvio padrão e variância (ou intervalos de confiança).

Encerrando, a aplicação em protótipos completos de jogos pode ampliar a compreensão sobre o impacto dessas abordagens na experiência do usuário e na performance do sistema.

Declarações complementares

Contribuições dos autores

Conceitualização, Software, Investigação, Curadoria de dados, Redação – rascunho original; M.B.V; Supervisão, Redação – revisão e edição; M.D.L. Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de dados e materiais

O software implementado para este trabalho, na forma de um projeto feito com a Unity, bem como os conjuntos de dados produzidos e analisados neste estudo (incluindo os dados brutos dos experimentos), estão disponíveis no repositório: <https://github.com/apecuca/MPC-Animal-Simulation>.

Outras informações relevantes

Foram utilizadas ferramentas de IA generativa de forma pontual, principalmente como apoio de revisão. No texto, elas serviram para conferir sintaxe e coerência, sugerir melhorias de clareza e reduzir ambiguidades em alguns trechos. No código, o uso se limitou a revisar partes específicas de lógica e a auxiliar na redação de comentários e documentação diretamente no código-fonte. Todas as sugestões foram conferidas e ajustadas pelos autores, que assumem integral responsabilidade pelo conteúdo final do artigo e pela correção técnica da implementação.

Referências

- Adeniyi, A. E., Brahma, B., Adebisi, M. O., Awotunde, J. B., Jimoh, R. G., Olasinde, E., and Bandyopadhyay, A. (2024). Development of two dimension (2d) game engine with finite state machine (fsm) based artificial intelligence (ai) subsystem. *Procedia Computer Science*, 235:2996–3006. DOI: 10.1016/j.procs.2024.04.283.
- Gil, A. C. (2017). *Como Elaborar Projetos de Pesquisa*. Atlas, São Paulo, SP, Brasil, 6 edition.
- Hu, W., Zhang, Q., and Mao, Y. (2011). Component-based hierarchical state machine — a reusable and flexible game ai technology. *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, 2:319–324. DOI: 10.1109/ITAIC.2011.6030340.
- Jagdale, D. (2021). Finite state machine in game development. *International Journal of Advanced Research in Science, Communication and Technology*, 10(1):384–390. DOI: 10.48175/ijarsct-2062.
- Maestre, J., Pena, D., and Camacho, E. (2011). Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32:153 – 176. DOI: 10.1109/CDC.2009.5400588.
- Martins, G., de Freitas, R., and Gadelha, B. (2023). Collaborative crowd games exploring participatory sensing and intelligent artificial life simulation techniques. *Revista Eletrônica de Iniciação Científica em Computação*, 21(1):54–63. DOI: 10.5753/reic.2023.2706.
- Naik, H., Bastien, R., Navab, N., and Couzin, I. D. (2020). Animals in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):2073–2083. DOI: 10.48550/arXiv.1912.12763.
- Negenborn, R. R., Schutter, B. D., and Hellendoorn, J. (2004). Multi-agent model predictive control: A survey. *ArXiv (Cornell University)*, abs/0908.1076. DOI: 10.48550/arXiv.0908.1076.
- Nugraha, C., Purnamasari, A., Bahtiar, A., and Tohidi, E. (2025). Implementation of finite state machine on npc to improve game productivity. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4:1673–1677. DOI: 10.59934/jaiea.v4i3.982.
- Saberi Moghadam, S., Khodadad, S., and Khazaeinezhad, V. (2019). An algorithmic model of decision making in the human brain. *Basic and Clinical Neuroscience Journal*, 10:443–450. DOI: 10.32598/bcn.9.10.395.
- Sani, M., Robu, B., and Hably, A. (2021). Pursuit-evasion games based on game-theoretic and model predictive control algorithms. In *2021 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–6. DOI: 10.1109/ICCAD52417.2021.9638775.
- Sugahara, L. E. and Lopes, M. D. (2021). Simulação de comportamento humano. In *Anais do Computer on the Beach*, volume 12, pages 347–354. DOI: 10.14210/cotb.v12.p347-354.
- Unity Technologies (2025). Unity 6.0 documentation. Acessado em: 06 nov. 2025.
- Waltham, M. and Moodley, D. (2016). An analysis of artificial intelligence techniques in multiplayer online battle arena game environments. In *Proceedings of the Annual*

Conference of the South African Institute of Computer Scientists and Information Technologists, SAICSIT '16, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2987491.2987513.

Wawrzyński, P., Arabas, J., and Cichosz, P. (2008). Predictive control for artificial intelligence in computer games. In Rutkowski, L., Tadeusiewicz, R., Zadeh, L. A., and Zurada, J. M., editors, *Artificial Intelligence and Soft Computing – ICAISC 2008*, pages 1137–1148, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-69731-2_107.

Wazlawick, R. S. (2009). *Metodologia de Pesquisa em Ciência da Computação*. Elsevier, Rio de Janeiro, RJ, Brasil. Selo Campus.