

# Simulação de Comportamento Humano

## Modelo de Personalidade Artificial para Personagens Não Jogáveis

Leonardo Enzo Sugahara

Marcelo Dornbusch Lopes

Universidade do Vale do Itajaí - UNIVALI

Florianópolis, Santa Catarina, Brasil

leonardosugahra@edu.univali.br

marcelo@univali.br

### RESUMO

Each new generation, games present physics and ambiance closer to their inspirations in the external world (non-virtual reality). However, the non-playable human characters in these games sometimes do not simulate the attitudes that a person would normally perform in front of a certain situation. This case goes against the proposal of a physical and graphically realistic environment, with behaviors and decision making not appropriate to the circumstances presented. To solve the problem, it is proposed the development of a Human Behavior Simulation Model, planned to be used on non-playable characters that will have constant contact with the player. This model was created based on concepts of Psychology, which were used to develop a personality that will be implemented in a character. To test the efficiency of the proposed model, it was used in the development of Carlos, a character of a game prototype. Carlos went through a series of controlled situations, used as tests to validate the applicability of the model in non-playable characters. The result was a coherent simulation of human behavior, at the cost of complexity in the application of the characteristics, described by Carlos' personality, to the character inside the game.

### KEYWORDS

Artificial Intelligence, Digital Games

## 1 INTRODUÇÃO

De maneira generalizada, [1] define o termo jogo como uma atividade lúdica (que visa a sensação de prazer, o divertimento), em que os participantes se veem em uma realidade fictícia, onde interagem (entre si ou com o próprio mundo do jogo) para alcançar ao menos um objetivo com base nas regras impostas por essa realidade.

Esse mundo fictício (jogo) possui um planejamento, um design por trás. Segundo [3], Design é algo que se preocupa com o planejamento e com o desenvolvimento de tudo aquilo que é criado de maneira artificial, ou seja, do que é criado pelo homem.

Dessa forma, Design de Jogos é o processo de planejamento e desenvolvimento do conteúdo de um jogo, ou seja, é definir qual a essência (conceito) de um jogo, descrever quais serão os seus elementos/componentes, bem como definir quais as regras e os objetivos que serão impostos ao jogador [21].

Com a presença desses elementos, surge a necessidade de balanceá-los até que o jogo transmita a experiência desejada ao jogador. Além do balanceamento, existe a importância de colocar todo o conteúdo planejado em um documento, organizando as informações para um futuro uso no processo de gestão [34].

Todo jogo possui requisitos, que incluem os recursos básicos de arte, design e programação. Esses recursos variam de gênero para gênero e podem ser alterados conforme a necessidade de cada projeto, uma vez que precisam estar de acordo com o conceito do jogo em questão [5].

A Inteligência Artificial (IA), um recurso da computação presente nos jogos, segundo [31] é uma área que visa estudar e desenvolver entidades inteligentes, capazes de realizar ações (e.g. pensar e agir) similares à de um ser humano ou de maneira racional.

Um exemplo de aplicação da IA nos jogos são os personagens não jogáveis (*Non-Playable Characters* - NPC), controlados pelo próprio sistema de inteligência artificial do jogo [27]. Os NPCs podem ser personagens que ficam apenas andando pelo cenário, um simples inimigo ou até mesmo alguém com um papel fundamental na narrativa do jogo.

A cada nova geração, os jogos apresentam uma física e uma ambientação mais próximas de suas inspirações presentes no mundo externo (realidade não-virtual). Jogos com essas características precisam de uma Inteligência Artificial que entenda os aspectos de um ambiente complexo e que consiga trabalhá-los de maneira otimizada e eficiente [32].

Além dos ambientes complexos, os jogos atuais também possuem personagens que procuram simular, da maneira mais fiel possível, os comportamentos humanos. Tal simulação de comportamento necessita de um modelo prévio, que pode ser obtido por meio da interpretação de características (e.g. emoções) pertencentes a uma personalidade, para só então implementá-la em um NPC [39].

De acordo com [19], o planejamento do comportamento humano para NPCs é geralmente complicado. Ainda segundo os autores, isso ocorre devido à ausência de um modelo que sirva de referência para definir como o comportamento em questão deve ser projetado. Além disso, existe o ponto de qual técnica de IA é a mais indicada para o desenvolvimento do personagem, uma vez que a sua escolha varia de acordo com o papel do NPC.

No contexto discutido, surge o seguinte objetivo: definir um modelo de comportamento humano para NPCs. Tal modelo busca coerência no comportamento do personagem perante as situações no mundo do jogo. Caso seja definido e implementado corretamente, o modelo resultará em uma maior imersão para o jogador durante o seu contato com os NPCs que possuem o modelo implementado.

Com o intuito de testar a proposta acima, foi desenvolvido um protótipo de jogo. Nele, existe um NPC com comportamento moldado a partir do modelo mencionado. Esse NPC possui um contato constante com o jogador, tornando mais dinâmica a validação do

objetivo proposto pelo modelo, justificando o desenvolvimento do protótipo mencionado.

No decorrer deste documento ainda são abordados os seguintes assuntos, todos eles com a sua própria seção apresentada em parêntesis: metodologias aplicadas na pesquisa (2), revisão bibliográfica sobre IA e conceitos de Psicologia (3), apresentação do Modelo de Simulação de Comportamento Humano desenvolvido (4), a aplicação do modelo em um NPC (5), além de uma conclusão sobre as informações contidas nesse artigo (6).

## 2 METODOLOGIA

A pesquisa contida neste documento partiu do problema gerado pela ineficiência no papel de NPCs que procuram simular o comportamento humano em jogos de caráter realista. Com o intuito de solucionar esse problema, foi definido um modelo de comportamento humano baseado em Personalidade Artificial.

Para avaliar se a solução proposta é ou não válida para resolver o problema apresentado, um protótipo de jogo foi desenvolvido. Nesse protótipo existe um NPC, no qual foi implementado o modelo de comportamento humano mencionado anteriormente.

De acordo com o exposto acima, segundo [20], a pesquisa pode ser classificada como uma aplicação do método hipotético-dedutivo, uma vez que ela contém: um problema (os NPCs ineficientes), uma hipótese (o modelo de Simulação de Comportamento Humano) e uma avaliação (o protótipo) para a solução proposta.

Quanto à natureza dessa pesquisa, segundo [8] é possível classificá-la como pesquisa aplicada, uma vez que os métodos e conceitos pesquisados foram utilizados em uma aplicação prática, nesse caso, o protótipo de jogo desenvolvido para testar o modelo definido.

Com base no objetivo de definir um modelo de comportamento humano para personagens não jogáveis, o artigo possui um nível de pesquisa exploratória, uma vez que conceitos e teorias precisaram ser pesquisados e esclarecidos para auxiliar na construção de uma hipótese pesquisável.

O nível de pesquisa deste documento (exploratório) pode ser definido, segundo [14], como um delineamento da pesquisa bibliográfica, sendo justificado pelo fato da existência de um levantamento bibliográfico prévio. Levantamento esse utilizado para refinar a hipótese de pesquisa: o uso do conceito de Personalidade Artificial para formular um modelo de comportamento humano.

## 3 REVISÃO BIBLIOGRÁFICA

Esta seção tem como objetivo apresentar uma breve descrição das técnicas de IA utilizadas no desenvolvimento de personagens não jogáveis. Assim sendo, são dispostos conceitos sobre algumas das técnicas implementadas em NPCs. Além disso, foi adicionada uma breve apresentação sobre a expressão Agentes Inteligentes, bem como a descrição de alguns modelos utilizados para simular o comportamento humano em Agentes.

### 3.1 Máquina de Estados Finitos

Máquina de Estado Finito (MEF) é uma técnica muito comum em jogos, baseada em estados e transições, cujo apenas um estado pode ser selecionado por vez [22]. Esses estados e transições são situações que podem vir a acontecer no jogo, portanto elas devem

ser antecipadas pelos desenvolvedores para serem adicionadas ao diagrama de estados [6].

Uma MEF simples possui 4 componentes básicos, são eles: os estados (onde ficam armazenadas as instruções de como o componente em questão deverá agir), as transições (são as relações entre os estados), as regras (delimitam um requisito para ocorrer a transição de um estado para outro) e os eventos (são acionados para verificar se a condição proposta pela regra em questão foi respeitada) [2].

A aplicação de uma Máquina de Estados, adaptado de [15], pode ocorrer em um NPC que interpreta um guarda cujo papel é apenas patrulhar uma área. Caso algo desconhecido seja avistado, o personagem não jogável irá persegui-lo, a fim de verificar do que se trata. Se o inimigo sair do campo de visão, o guarda voltará a patrulhar.

Do contrário, ao se aproximar o suficiente e após validar que o alvo é um inimigo, o guarda irá atacá-lo. Caso o inimigo seja eliminado ou saia do campo de visão, o personagem voltará ao seu estado de patrulha. Se não, o guarda continuará a perseguição até que o inimigo seja atacado e eliminado. A Máquina de Estados Finitos utilizada no NPC está representada na Figura 1.

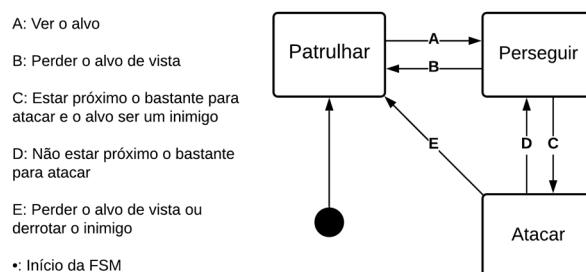


Figura 1: Máquina de Estados Finitos do Guarda

Quando uma MEF apresenta poucos estados, como exemplificado acima, a sua manutenção e a implementação de novos comportamentos tornam-se algo simples. Contudo, conforme a MEF vai crescendo, a situação se reverte: a quantidade de estados e transições que são adicionadas à máquina tornam cada vez mais difícil manter a sua funcionalidade [6].

Todavia, além de uma simples implementação, as MEFs também são intuitivas e requerem uma pequena quantidade de recurso computacional. Isso ocorre graças à separação das ações em estados únicos, divididos em sequências de operadores condicionais, que simplificam a visualização das ações enquanto facilitam a sua depuração [4].

### 3.2 Lógica Fuzzy

Ao contrário da lógica clássica (binária), a lógica Fuzzy (também conhecida como lógica difusa) busca trazer um maior intervalo de possibilidades para se trabalhar. Em outras palavras, o que antes era apenas ou 0 ou 1, ao usar a lógica Fuzzy tem-se a abertura para utilizar quaisquer valores contidos em um intervalo de dois números, como por exemplo entre os próprios 0 e 1 [2].

Quando a lógica difusa é implementada em uma MEF, obtém-se uma Máquina de Estados Finitos baseada em Fuzzy (MEFF). O

resultado dessa união é uma MEF que possui algum elemento difuso, ou seja, uma MEFF que apresenta a lógica de Fuzzy aplicada em suas transições e/ou em seus eventos [22]. Tal aspecto promove um fato interessante sobre a MEFF: ela pode ter mais de um estado ativo [17].

A aplicação de uma Máquina de Estados Finitos baseada em Fuzzy, assim como mencionada por [39], pode ocorrer em um NPC que tem como finalidade passar ao jogador uma sensação de imersão, através de seu comportamento. Personagens com esse propósito são comuns nos jogos focados em narrativa, cuja história é repleta de atores passíveis de interação com o jogador [33].

Com o objetivo de passar uma maior imersão ao jogador, a MEFF pode ser utilizada para definir quais serão os estados de comportamento desse personagem, bem como suas ações [39]. Ambas serão baseadas em condições específicas, que caso bem descritas e implementadas, garantirão uma transição de comportamento mais natural [28], já que o comportamento humano não se restringe a apenas um estado [24].

Contudo, implementações que utilizam a Lógica Fuzzy também possuem seus poréns. Se não forem planejados com cuidado, os sistemas que utilizam Fuzzy (e.g. que controlam personagens em um jogo) podem chegar a conter centenas de regras, que serão verificadas a todo o momento. Tal característica anula uma das vantagens presentes ao utilizar essa técnica: o baixo custo computacional [28].

### 3.3 Redes Neurais Artificiais

O corpo humano possui um sistema de células responsável por controlar as suas funções e os seus movimentos. Tal sistema é composto por neurônios, células capazes de transmitir informações para outras partes do corpo através de ramificações, que se conectam a outros neurônios. As conexões geradas por grupos de neurônios formam uma espécie de rede, conhecida como Rede Neural [7].

Baseadas nessas conexões e em como o cérebro humano resolve determinados problemas, as Redes Neurais Artificiais (RNA) são sistemas computacionais complexos adaptativos, capazes de aprender e de se adaptar para solucionar um problema da maneira mais eficiente possível. Tal eficiência é determinada por pesos, redefinidos pelo sistema quando uma solução ruim é encontrada [35].

Uma aplicação de Redes Neurais Artificiais nos NPCs pode ser observada naqueles personagens cujo objetivo é enfrentar os jogadores humanos. Em implementações como essa, as RNAs podem ser treinadas com dados obtidos a partir das decisões presentes nas jogadas desempenhas, no jogo em questão, por jogadores humanos [9, 19]. Um exemplo desse tipo de uso é em jogos First Person Shooter (FPS)<sup>1</sup>.

A consequência desse procedimento, no exemplo mencionado acima, são NPCs que podem errar tiros em algumas situações, focam os alvos dentro de seu raio de visão, além de simular um tempo de reação ao invés de atacar assim que um inimigo é detectado. Em outras palavras, o comportamento do NPC fica mais próximo ao de um humano, minimizando as sensações de estar enfrentando uma máquina [38].

Contudo, a qualidade da solução (saída) encontrada pela RNA é imprevisível [23]. Para amenizar esse problema, é possível utilizar uma maior quantidade de dados durante o treinamento da rede [26]. Isso fará com que a IA tenha mais informações para analisar, podendo guiar o aprendizado da máquina na direção da solução desejada.

### 3.4 Agentes Inteligentes

O termo Agente refere-se a algo (real ou artificial) que utiliza sensores para perceber o ambiente ao seu redor, além de realizar ações sobre esse mesmo ambiente através de atuadores. Um exemplo dessa ideia é o próprio ser humano: o homem possui os cinco sentidos (dentre outros) para identificar o ambiente onde se encontra, mas também pernas, braços e boca (dentre outros) que servem como atuadores [31].

Em ambientes artificiais (e.g. jogos digitais), a expressão Agente Inteligente é definida como um programa que percebe o ambiente onde está inserido e age sobre ele. Além disso, esse Agente também possui a liberdade (mesmo que mínima) para escolher quais são as melhores decisões, que como consequência irão ajudá-lo a alcançar seus objetivos [37].

Os Agentes Inteligentes podem ser encontrados, por exemplo, em jogos de corrida como *Colin McRae Rally 2*<sup>2</sup> e na série *Forza*<sup>3</sup>, cujo uma RNA é utilizada para controlar os carros adversários [6]. Uma outra aplicação são em FPS como o jogo *Unreal*<sup>4</sup>, que possuem uma MEFF implementada para controlar o comportamento dos NPCs inimigos [28].

### 3.5 Comportamento Humano em Agentes Inteligentes

Em jogos digitais que possuem NPCs, a imersão do jogador está ligada ao fato dos personagens passarem a sensação de que são quem eles representam. Para NPCs cujo papel é interpretar um ser humano, existem modelos que têm como principal objetivo simular comportamentos, emoções e/ou personalidades, cada um com seu próprio foco, mas com um único objetivo: tornar o comportamento do personagem mais humano [25].

No modelo de emoções documentado por [10], a proposta é que o agente crie relações com outros NPCs ou até mesmo com o próprio jogador. Essa conexão acontece através de conceitos de inteligência emocional (e.g. entender, mas também ser capaz de identificar emoções em si mesmo e em outros indivíduos) que foram utilizados para elaborar o modelo em questão.

Quanto ao modelo de personalidade documentado por [11], a ideia é que o personagem demonstre, através de seu comportamento, um Arquétipo de Personalidade (AP). Jung [16] define Arquétipo como uma forma, um molde do comportamento de um indivíduo, baseado em crenças herdadas geneticamente de seus ancestrais (e.g. pais, avós).

Já a Personalidade pode ser definida como um conjunto de fatores (e.g. emocionalidade, sociabilidade) que tornam uma pessoa única, derivados da experiência individual e comportamental de

<sup>1</sup>O gênero *First Person Shooter* engloba jogos de ação que colocam o jogador na visão do personagem controlado, ou seja, a câmera normalmente é posicionada na altura dos olhos do ator controlado. O foco desse estilo de jogo é em conflitos armados [29]

<sup>2</sup>*Colin McRae Rally 2*, Codemasters, 2000

<sup>3</sup>Série *Forza*, Xbox Game Studios, 2005 – 2018

<sup>4</sup>*Unreal*, Epic Games, 1998

cada indivíduo [13]. Assim sendo, esse tipo de modelo define o comportamento do personagem a partir das características presentes em seu Arquétipo.

Além do Arquétipo, também é possível categorizar as características únicas de cada indivíduo através dos Traços de Personalidade (TP) [13]. Os TPs são as diferenças presentes em cada ser, que contribuem para modelar o seu próprio comportamento [12]. Essas diferenças podem ser definidas em Traços Internos (TI), Traços Externos (TE), Gostos (G), Medos e Fobias (MF), dentre outros [18].

#### 4 MODELO DE SIMULAÇÃO DE COMPORTAMENTO HUMANO

O Modelo de Simulação de Comportamento Humano (MSCH) trata-se de um molde planejado para personagens não jogáveis, que tem como objetivo simular as ações de um ser humano. Nesse modelo, a ideia é organizar e planejar quais serão os comportamentos de um NPC, baseados em uma metodologia descrita pelo molde. Além disso, o MSCH também define qual técnica de IA (e.g. MEF ou RNA) será utilizada no desenvolvimento do personagem.

Planejado para tornar o comportamento dos personagens humanos próximo ao retratado no mundo não-virtual, o MSCH foi desenvolvido para ser utilizado em NPCs que além de ter um contato constante com o jogador, também atuem diretamente nos acontecimentos do jogo (e.g. participar de uma batalha ou abandoná-la, influenciar o comportamento de outros personagens, danificar propositalmente um abrigo).

Ter a propriedade para alterar os acontecimentos durante o jogo impacta diretamente a experiência de quem joga, portanto é preciso implementar indicadores encarregados de informar sobre uma possível alteração no comportamento do personagem, dando ao jogador uma oportunidade de se preparar para o acontecimento.

Visando essa coerência no comportamento dos NPCs, o conceito presente no modelo de Personalidade Artificial foi adotado para o desenvolvimento do MSCH. Isso ocorreu devido à presença dos Arquétipos de Personalidade, que descrevem de maneira objetiva o comportamento do personagem. Além dos APs, também é possível utilizar Traços de Personalidade para formular a personalidade dos personagens, já que um grupo de TPs compõem um AP [12].

A objetividade proposta pela personalidade auxilia no planejamento das ações que o personagem pode realizar, consequentemente impactando na experiência do jogador. Outro ponto interessante é a oportunidade de unir a história do NPC com o seu AP, ligando o seu comportamento com as experiências adquiridas por esse personagem ao longo de sua história. Essa poderosa junção pode resultar em uma conexão com o jogador, tornando sua experiência mais imersiva [11].

Com base nos fatos acima, o MSCH inicia-se em duas partes focadas no planejamento do NPC: Etapa 1 (E1) e Etapa 2 (E2). Em E1, o foco é no design da personalidade do NPC, ou seja, a criação do personagem será a principal atividade. Elementos como história, MF, bem como do que o NPC gosta ou não, vestimentas e aparência do personagem, dentre outros aspectos relacionados, são definidos nessa etapa.

Visando o desenvolvimento de um personagem fictício convincente, recomenda-se a descrição de TPs, como:

- (1) TI - São traços comportamentais adquiridos com experiências de vida ou herdados geneticamente. Exemplos: observador (sempre atento aos possíveis riscos/oportunidades), pensativo (utiliza seu conhecimento e experiência para tomar melhores decisões);
- (2) TE - Contêm características comportamentais baseadas em como o personagem em questão é visto pelos demais. Exemplos: lacônico (poucas palavras, direto ao ponto), hábil (é um dos melhores no que faz, sempre encontra as soluções);
- (3) G - É descrito tudo (e.g. objetos, atividades/ações, seres vivos) que deixa o personagem feliz e/ou triste. Exemplos: Não gostar de frutas ácidas, amar carne e adorar nadar;
- (4) MF - São listados todos os medos e fobias do NPC. Exemplos: fracasso (não conseguir concluir um objetivo, uma meta ou falhar em cumprir uma promessa/missão), hemofobia (ver/entrar em contato com sangue). Além de elencar os elementos que aterrorizam o personagem, também é preciso definir a influência que cada um possui sobre ele. Os valores dessa suscetibilidade ao medo serão utilizados futuramente no modelo e são estabelecidos como: 0.5 para baixa, 1 para mediana e 2 para alta;
- (5) História - É a história do personagem. Nela, podem ser encaixados todos os elementos descritos em 1, 2, 3 e 4, fazendo com que o jogador possa se identificar, de maneira profunda, com o personagem.

Ao juntar os pontos mencionados acima (do 1 ao 5), surge uma nova personalidade que irá influenciar em todo o comportamento do personagem.

Já em E2, o trabalho consiste em planejar a futura aplicação da personalidade desenvolvida em E1. O primeiro passo é definir quais serão os Estados Emocionais (EE) (e.g. medo e calma) que o personagem pode atingir. Após, são estabelecidas as Ações Básicas (AB) (e.g. atacar e andar) que o NPC pode executar, além dos Atos Exclusivos (AE) oriundos de sua personalidade.

Os AEs são ações específicas de cada personagem, que têm como objetivo realçar a sua personalidade. Esses atos podem ser algo que o NPC goste de fazer (e.g. ler, explorar e analisar o terreno) ou então uma reação (leve ou extrema) a algo (e.g. pulos de felicidade em cenários alegres, paralisia do corpo em situações de medo).

A partir das informações reunidas em E1 e E2, têm-se um modelo organizado sobre o comportamento e a personalidade do personagem. Em outras palavras, as suas ações, reações e EEs já podem ser incumbidos ao NPC através de uma IA. Essa aplicação é descrita na Etapa 3 (E3), dividida em duas linhas: a Parte 1 (P1) e a Parte 2 (P2).

Em P1, é implementada ao personagem uma MEF, cujos estados são as ABs e os AEs definidos em E2. As transições, assim como as regras, variam de acordo com as situações que podem ser apresentadas ao decorrer do jogo, portanto é possível ter regras e transições exclusivas para cada projeto.

Por sua vez, em P2 é utilizada a técnica de MEF para implementar os EE, definidos em E2, no personagem. Visto que o comportamento humano não se restringe a apenas um estado, foi desenvolvida uma adaptação para os estados dessa MEF. Esse ajuste trata-se da implementação de dois tipos de estados: os majoritários e os minoritários.



O primeiro faz jus aos tipos de comportamento que possuem prevalência no personagem, ou seja, nesse grupo são dispostos os EEs predominantes no comportamento do NPC. Para indicar ao jogador qual é o atual EE majoritário do personagem, podem ser utilizadas variadas animações para cada AB do NPC. Um exemplo seria a seguinte variação da AB andar: andar normalmente e andar expressando tristeza.

Quanto aos minoritários, seu papel é de atuar como Estados de Transição (ET) para os majoritários. Assim sendo, o grupo de EEs minoritário não influencia tanto no comportamento do personagem quanto o grupo majoritário. Dessa maneira, para indicar a troca de estados, os ETs fazem uma alteração nos status (e.g. velocidade de ataque, raio de visão) do NPC. A Figura 2 apresenta as etapas que compõem o MSCH.

A ideia por trás da implementação dos estados majoritários e minoritários são as transições no comportamento do NPC. Para isso, os ETs devem alterar o status de maneira que as animações presentes nos EEs majoritários justifiquem o comportamento do NPC. Um exemplo seria o NPC estar com medo e desmotivado, apresentando uma animação de andar que expressa tristeza, além de uma velocidade de movimento abaixo do padrão.

Seguindo com o exemplo acima, o mesmo personagem acaba encontrando um objeto que lhe traz boas lembranças e isso o ajuda a recuperar a motivação. Antes da troca de estado majoritário acontecer, um ET entra em ação. Nesse estado, o aumento progressivo na velocidade de movimento acontece até o valor estabelecido no próximo EE minoritário, suavizando a substituição de EEs.

Contudo, apenas alterar gradualmente o valor presente nos status não irá tornar a transição tão efetiva. Para contornar o problema, é preciso que as animações também acompanhem o mesmo ritmo de troca, ou seja, elas também necessitam passar por um processo de transição. Portanto, é fundamental a presença de animações que expressem nitidamente os EE planejados em E2.

Dessa maneira, o personagem sempre estará em dois estados: um majoritário, representando o EE predominante e um minoritário, indicando o próximo EE do NPC. Para controlar essas transições entre os EEs, foi desenvolvida a variável Sanidade (S), que funciona como regra principal para validar as trocas de estados. Para calcular S, são utilizados três parâmetros: Felicidade (F), Medo (M) e Peso do Medo (PM).

O elemento F é definido pela porcentagem de felicidade do personagem, alterada sempre ao entrar em contato (visual ou não) com algo que o deixa feliz ou triste. Já a variável M, também em porcentagem, muda de acordo com a exposição (e.g. estar dentro do campo de visão e/ou próximo) do personagem ao que lhe causa medo. Por fim existe o PM, modificado durante o contato do NPC com algum fobia.

Para alterar o valor das variáveis F, M e PM descritas acima, o NPC precisa perceber o elemento em questão, seja através de seu campo de visão/detecção ou de outra maneira. Dessa forma, as peças responsáveis por alterar F, M e PM irão variar de personagem para personagem (pois precisam ser descritas em sua personalidade), bem como de jogo para jogo (já que os projetos podem ter personagens e elementos únicos).

Uma vez que S, F e M são representadas em porcentagem, neste modelo seus valores ficam restritos a 100%, ou seja, essas variáveis podem ser encontradas no intervalo de [0, 100]. Já PM é um valor

dependente das Fobias (FB) e da Suscetibilidade ao Medo (SM), ambas descritas em E1 durante o desenvolvimento da personalidade do personagem. O valor mínimo de PM é controlado por SM, enquanto suas FB irão alterar PM durante o jogo. Portanto, seu intervalo é descrito por  $[SM, FB]$ .

Neste modelo de comportamento, F é um parâmetro utilizado para suprir a imprevisibilidade de M. Contudo, caso ele esteja em contato com uma de suas fobias, o valor de M deve ser maior. Para isso ocorrer, PM é alterado de acordo com o impacto da fobia em questão. Assim sendo, o valor de S pode ser descrito pela seguinte expressão:  $S = F - (M * PM)$ .

## 5 APLICAÇÃO MSCH

Para validar o MSCH, um protótipo foi desenvolvido utilizando a *Unity Engine* e a linguagem C#. O projeto trata-se de um jogo 3D cujo gênero é de sobrevivência<sup>5</sup> com elementos de RTS<sup>6</sup>, em que o jogador precisa explorar o cenário com um grupo de NPCs aliados para permanecerem vivos. Um dos personagens desse grupo recebeu uma personalidade artificial oriunda do MSCH.

No protótipo foi criado um tutorial para mostrar as mecânicas ao jogador. Nessa fase, faz-se o uso de um dos NPCs para retratar uma série de possíveis acontecimentos, que podem alterar o seu comportamento, baseado em uma série de variáveis descritas no MSCH. Para saber em qual estado se encontra o NPC, pode-se observar as suas animações, bem como o valor de sua sanidade.

Além disso, ainda é possível clicar no personagem, que resultará na aparição de uma interface no canto superior esquerdo da tela. Nela, estão contidas informações acerca do estado emocional do NPC, bem como os seus status (e.g. dano, velocidade de movimento). Portanto, o experimento ocorreu da seguinte maneira: o NPC com o MSCH foi exposto a situações controladas e seu comportamento foi observado.

O NPC selecionado foi Carlos, um pesquisador do exército. Apaixonado pelo poder do conhecimento, ele possui cerca de 40 anos de idade e já passou por diversas experiências ruins, que resultaram em um traço de personalidade lacônico. Contudo, Carlos ainda é uma pessoa genuína, que luta pela igualdade e se sente bem ao ajudar quem precisa. Os demais dados da personalidade de Carlos encontram-se na Tabela 1.

**Tabela 1: Elementos da personalidade de Carlos**

TI	TE	Gosta	Não gosta	FM
Observador	Hábil	Animais	Cerveja	Fracasso
Curioso	Lacônico	Plantas	Frutas	-
Restrito	Estudioso	Ferramentas	-	-
Corajoso	Bom	Explorar	-	-

<sup>5</sup>Subgênero de Aventura, os jogos de sobrevivência têm como objetivo expor o jogador a situações que o obrigam a coletar recursos e explorar o cenário para conseguir superar os desafios propostos pelo jogo [30]

<sup>6</sup>Jogos de Estratégia em Tempo Real (*Real Time Strategy* – RTS), são um subgênero de Estratégia, que têm como principal característica a ausência de turnos para o planejamento das ações [36]

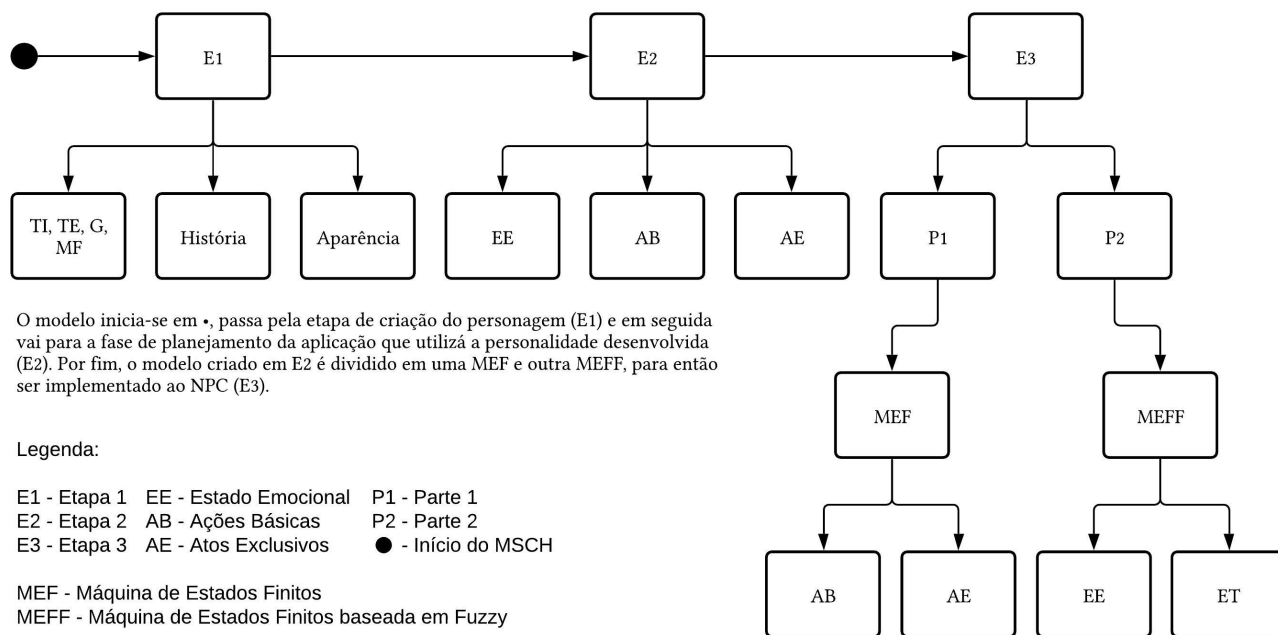


Figura 2: Etapas do MSCH

Para criar a tabela com os dados obtidos, foram utilizadas siglas referentes aos status do NPC. São elas: Dano de Ataque (DA), Velocidade de Ataque (VA), Velocidade de Movimento (VM), Raio de Visão (RV), Sanidade (S) e Estado Emocional Majoritário (EEM).

A Tabela 2 contém todos os dados obtidos e é organizada da seguinte maneira: Caso (a situação simulada no teste), Comportamento (ações/comportamentos previstos para acontecer com o personagem), Status (sigla do status que sofreu alteração), Inicial (valor do status no início da simulação), Final (valor do status no final da simulação).

Além do exposto acima, todas as entradas de dados vieram da simulação de percepção no sentido visão do personagem Carlos, ou seja, para serem detectados os elementos precisaram estar dentro de um campo definido com sendo seu raio de visão. Portanto não foram utilizados outros métodos de simulação de percepção (e.g. audição) como entrada de dados.

Com base nos dados coletados a partir das situações apresentadas, os status finais e o comportamento de Carlos foram coerentes com o proposto pela sua personalidade. Uma vez que, apreensivo em fracassar ao proteger alguém, Carlos sente medo quando um dos companheiros é atacado. Isso fornece a ele um bônus de 2 pontos em VM e 1 em VA, para ajudar seu colega o mais rápido possível.

No entanto, a situação em questão causou um leve amedrontamento em Carlos, aumentando a variável Medo e conseqüentemente diminuindo sua Sanidade. Ao cair, a Sanidade deixou o EE minoritário do personagem mais próxima do EEM do Medo. Isso fez com que o NPC tivesse sua VM restrita em quatro pontos, limitando sua velocidade máxima e reduzindo o bônus em VM fornecido para ajudar seus companheiros.

Tabela 2: Resultados dos Testes

Caso	Comportamento	Status	Inicial	Final
Companheiro ser atacado	Ajudar assim que possível.	VM	3	4
	Aumento em VM e VA.	VA	1	2
	Medo subir.	S	45	44.2
Companheiro ser eliminado	Ficar abalado.	S	44.7	16.2
	Encarar o corpo.	RV	30	15
	Felicidade cair.	EEM	Neutro	Medo
Salvar um companheiro	Socorrer o companheiro.	S	44.2	53.6
	Felicidade subir.			
Ver um coelho	Ficar feliz ao ver algo que gosta.	S	45	45.7
	Felicidade subir.			
Ver 3 inimigos próximos	Não se abalar.	DA	10	15
	Subir DA e VA.	VA	1	2

Ao detectar um companheiro sendo eliminado em sua presença, Carlos fica muito abalado (pois fracassou em proteger alguém). Como resultado sua Felicidade caiu drasticamente, diminuindo sua

Sanidade e consequentemente alterando seu EEM de Neutro para Medo. Essa mudança afetou o seu RV, adaptado ao seu EE atual. Além disso, o personagem ainda fica parado por alguns segundos, encarrando o corpo do colega antes de realizar outra ação.

Nas duas situações seguintes, a primeira em que Carlos salva um companheiro e a segunda retratando quando o personagem encontra um coelho, o NPC em questão fica aliviado e feliz ao fazer/presenciar algo que o deixa tranquilo. Isso acontece pois Carlos tem medo de fracassar, seja em proteger alguém ou ao cometer um erro grave e, quando triunfa sobre a chance de derrota, ele se sente bem. Além disso, Carlos gosta muito de animais. Portanto, em ambas as situações, sua Felicidade e consequentemente sua Sanidade aumentam.

Já na última situação, visto que Carlos possui um Traço Interno de coragem, ao avistar três inimigos seu medo não se alterou. Ao invés disso, seu o DA e VA aumentaram para proteger seus companheiros. O efeito apenas aconteceu enquanto os três inimigos estão em seu raio de visão. Quando um deles saiu ou foi derrotado, o bônus terminou.

Contudo, durante os testes iniciais, ocorreram situações em que os NPCs (com e sem o MSCH) não detectavam, em seu campo de visão, a presença do inimigo. Isso aconteceu porque a lista dos alvos localizados era atualizada com um atraso (proposital) de 0.2 segundos. O resultado foi uma percepção atrasada que, até ser corrigida, atrapalhou os primeiros testes, pois acabou alterando os dados de entrada e consequentemente, a ação realizada pelo NPC.

## 6 CONCLUSÃO

O MSCH foi planejado para NPCs que tenham um constante contato com o jogador. Partindo disso, é necessário a presença de elementos (Estados de Transição, animações específicas para cada Estado Emocional) responsáveis por tornar o comportamento do personagem mais natural, próximo da realidade. Para tal, o MSCH foi baseado no conceito de Personalidade Artificial e utilizou a técnica de MEFF para implementar os EE da personalidade ao NPC.

Durantes os testes, os ETs e as animações exclusivas para cada EE foram essenciais para tornar o comportamento de Carlos, o NPC analisado na Seção 5, coerente com as situações simuladas e com as características de sua Personalidade Artificial. Contudo, uma vez que os acontecimentos apresentados nos testes foram avaliados separadamente, durante o jogo podem haver cenários em que Carlos se comporte de maneira diferente.

Isso pode acontecer devido à chance de algumas situações ocorrerem simultaneamente, já que os acontecimentos, durante o jogo, dependem de uma série de fatores (e.g. ações dos personagens, regras do jogo), podendo muitas vezes sair do que foi planejado pelos desenvolvedores. Portanto, é interessante pensar em uma possível combinação dos bônus oferecidos pelo MSCH durante o balanceamento do NPC.

Por fim, aplicar as características descritas no MSCH de Carlos ao NPC presente no protótipo foi uma tarefa complexa e trabalhosa. O motivo é simples: a união de elementos como criação de personagem, animações exclusivas para cada EE, além de componentes (e.g. itens, NPCs) únicos para cada jogo, dentre outros, em um modelo comportamental para Agentes Inteligentes é algo que necessita de um árduo planejamento.

Caso algum dos integrantes dessa união seja adicionado ao modelo sem uma organização prévia, o resultado pode ser diferente do esperado. Em outras palavras, ao invés de propor uma maior imersão ao jogador, o modelo desenvolvido e aplicado sem um planejamento poderá resultar em um personagem com atitudes estranhas, incoerentes com a situação retratada no jogo.

Como sugestão de trabalhos futuros, pretende-se adicionar mais opções para entrada de dados, como uma simulação de percepção envolvendo o sentido audição, por exemplo. Além disso, o protótipo será evoluído para testar e avaliar o MSCH em um ambiente mais próximo de um jogo finalizado. Ambas as implementações têm por objetivo aperfeiçoar o modelo, tornando-o mais viável.

## REFERÊNCIAS

- [1] Ernest Adams. 2014. *Fundamentals of Game Design* (3rd ed.). New Riders Publishing, USA.
- [2] Ray Barrera, Aung Sithu Kyaw, Clifford Peters, and Thet Naing Swe. 2015. *Unity AI Game Programming - Second Edition*. Community Experience Distilled, Vol. Second edition. Packt Publishing. <https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1071382&lang=pt-br&site=ehost-live>
- [3] Richard Buchanan. 1992. Wicked Problems in Design Thinking. *Design Issues* 8, 2 (1992), 5–21. <http://www.jstor.org/stable/1511637>
- [4] Mat Buckland. 2004. *AI game programming by example*. Wordware.
- [5] Heather Maxwell Chandler. 2012. *Manual De Produção De Jogos Digitais* (2 ed.). Bookman.
- [6] Ricardo Cherobin. 2014. *Aplicação do modelo da cadeia alimentar juntamente com algoritmos genéticos para criação de NPCs adaptativos*. [manuscrito]. <https://search.ebscohost.com/login.aspx?direct=true&db=SIB.208956&authtype=shib&lang=pt-br&site=eds-live&scope=site>
- [7] Dennis Kerr Coelho. 2005. *Sistema Neural Para Previsão De Tempo De Perfuração De Poços De Petróleo*. Master's thesis. Universidade Federal de Santa Catarina. <https://repositorio.ufsc.br/bitstream/handle/123456789/102655/222388.pdf?sequence=1&isAllowed=y>
- [8] Edna Lucia Da Silva and Estera Muszkat Menezes. 2005. *Metodologia da Pesquisa e Elaboração de Dissertação* (4 ed.). UFSC.
- [9] D. de Almeida Rocha and J. Cesar Duarte. 2019. Simulating Human Behaviour in Games using Machine Learning. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. 163–172. <https://doi.org/10.1109/SBGames.2019.00030>
- [10] João Dias and Ana Paiva. 2013. I Want to Be Your Friend: Establishing Relations with Emotionally Intelligent Agents. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (St. Paul, MN, USA) (AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 777–784.
- [11] Benjamin Ellinger and Steve Rabin. 2008. *Artificial Personality: A Personal Approach to AI*. Cengage Learning, 17–26.
- [12] Jess Feist, Gregory J. Feist, and Tomi-Ann Roberts. 2015. *Teorias da Personalidade* (8 ed.). Amgh.
- [13] Mirna Torres Figueiró, José Carlos Souza, Lucy Nunes Ratier Martins, Lucas Rasi Cunha Leite, Juliana Martimbianco Ziliotto, and Mayara Mendes Bacha. 2010. Traços de personalidade de estudantes de Psicologia. *Psicologo informacao* 14 (10 2010), 13 – 28. [http://pepsic.bvsalud.org/scielo.php?script=sci\\_arttext&pid=S1415-88092010000100002&nrm=iso](http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1415-88092010000100002&nrm=iso)
- [14] Antonio Carlos Gil. 1995. *Métodos e técnicas de pesquisa social* (4 ed.). Atlas.
- [15] David Graham. 2017. *A Reusable, Light-Weight Finite-State Machine*. 159–166.
- [16] Carl G. Jung. 2002. *Os arquétipos e o inconsciente coletivo*. Vozes.
- [17] C. Langensiepen, A. Lotfi, and S. Puteh. 2014. Activities recognition and worker profiling in the intelligent office environment using a fuzzy finite state machine. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 873–880. <https://doi.org/10.1109/FUZZ-IEEE.2014.6891825>
- [18] Howard Lauther. 2004. *Creating characters: a writers reference to the personality traits that bring fictional people to life*. McFarland.
- [19] Joan Marc Llargues Asensio, Juan Peralta, Raul Arrabales, Manuel Gonzalez Bedia, Paulo Cortez, and Antonio Lopez Peña. 2014. Artificial Intelligence Approaches for the Generation and Assessment of Believable Human-like Behaviour in Virtual Characters. *Expert Syst. Appl.* 41, 16 (Nov. 2014), 7281–7290. <https://doi.org/10.1016/j.eswa.2014.05.004>
- [20] Marina de Andrade Marconi and Eva Maria Lakatos. 2003. *Fundamentos de Metodologia Científica* (5 ed.). Atlas.
- [21] Vicente Martin MASTROCOLA. 2016. *Game design: modelos de negócios e processos criativos: um trajeto do protótipo ao jogo produzido*. Cengage Learning.
- [22] Ian Millington. 2019. *AI for games*. CRC Press.

- [23] M. Mishra and M. Srivastava. 2014. A view of Artificial Neural Network. In *2014 International Conference on Advances in Engineering Technology Research (ICAETR - 2014)*, 1–3. <https://doi.org/10.1109/ICAETR.2014.7012785>
- [24] Gadelhag Mohamed, Ahmad Lotfi, and Amir Pourabdollah. 2020. Enhanced fuzzy finite state machine for human activity modelling and recognition. *Journal of Ambient Intelligence and Humanized Computing* 11 (2020). <https://doi.org/10.1007/s12652-020-01917-z>
- [25] John Mooney and Jan M. Allbeck. 2014. Rethinking NPC Intelligence: A New Reputation System. In *Proceedings of the Seventh International Conference on Motion in Games (Playa Vista, California) (MIG '14)*. Association for Computing Machinery, New York, NY, USA, 55–60. <https://doi.org/10.1145/2668084.2668091>
- [26] Henry Navarro and L. Bennun. 2014. Descriptive examples of the limitations of Artificial Neural Networks applied to the analysis of independent stochastic data. *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY* 5 (05 2014), 40–42.
- [27] Jeannie Novak. 2008. *Game development essentials: an introduction*. Delmar Cengage Learning.
- [28] M. Pirovano. 2012. The use of Fuzzy Logic for Artificial Intelligence in Games.
- [29] Steve Rabin. 2012. *Introdução ao desenvolvimento de games - Volume 1: Entendendo o universo dos jogos*. Cengage Learning.
- [30] Scott Rogers. 2013. *Level UP: um Guia Para o Design de Grandes Jogos* (1 ed.). Blucher.
- [31] Stuart Russel and Peter Norvig. 2013. *Inteligência Artificial*. Elsevier.
- [32] Firas Safadi, Raphael Fonteneau, and Damien Ernst. 2015. Artificial Intelligence in Video Games: Towards a Unified Framework. *Int. J. Comput. Games Technol.* 2015, Article 5 (Jan. 2015), 1 pages. <https://doi.org/10.1155/2015/271296>
- [33] Kevin Saunders and Jeannie Novak. 2013. *Game development essentials: game interface design*. Delmar Cengage Learning.
- [34] Jesse Schell. 2014. *The Art of Game Design: A Book of Lenses* (2nd ed.). A. K. Peters, Ltd., USA.
- [35] Daniel Shiffman. 2012. *The nature of code*. The Magic Book Projekt.
- [36] Xiong Shuo and Iida Hiroyuki. 2015. Attractiveness of Real Time Strategy Games. *Computer Science & Information Systems* 12, 4 (2015), 1217 – 1234. <https://search.ebscohost.com/login.aspx?direct=true&db=iih&AN=111524881&lang=pt-br&site=ehost-live>
- [37] B. Silverman, M. Johns, R. Weaver, K. O'Brien, and R. Silverman. 2002. Human Behavior Models for Game-Theoretic Agents: Case of Crowd Tipping.
- [38] B. Soni and P. Hingston. 2008. Bots trained to play like a human are more fun. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 363–369. <https://doi.org/10.1109/IJCNN.2008.4633818>
- [39] Zhen Liu. 2008. A personality model of virtual characters. In *2008 7th World Congress on Intelligent Control and Automation*. 2497–2500. <https://doi.org/10.1109/WCICA.2008.4593316>