

EMPREGO DE EMOÇÃO COMO UTILIDADE EM ÁRVORES DE COMPORTAMENTO

André Luís Souza Gonçalves¹, André Faria Ruaro²

Resumo: Desde sua concepção, os responsáveis pelo desenvolvimento de jogos eletrônicos buscam formas de trazer personagens não jogáveis capazes de demonstrar (ou fingir) comportamentos dinâmicos e inteligentes. Das formas populares de implementar esses comportamentos, sua grande maioria trabalha com gerenciamento de estados, tendo o automato finito como exemplo mais popular. As árvores comportamentais, apesar de se encontrar como uma alternativa superior, também possui suas limitações. O objetivo deste trabalho é implementar uma evolução dessas árvores com a utilização de abstrações de utilidade, representadas como necessidades e estados emocionais, de forma a permitir a reorganização das árvores comportamentais superando as dificuldades levantadas por Colledanchise e Ögren sobre a utilização de utilidade para o mesmo. O método utilizado será o desenvolvimento de um protótipo jogável capaz de demonstrar as técnicas utilizadas, suas forças e limitações, permitindo a criação de jogos eletrônicos com personagens capazes de demonstrar comportamentos inteligentes, mesmo sem relação claramente projetada entre agentes, trazendo uma jogabilidade dinâmica e lúdica. Os resultados obtidos suportam a viabilidade das técnicas descritas, assim como demonstra de forma clara seus pontos fortes e limitações. Apesar das limitações apresentadas e considerações necessárias para a introdução das técnicas descritas neste trabalho em um projeto, demonstramos de forma clara que, em projetos que buscam agentes com comportamento inteligente e possuem um foco nas relações entre entidades e comportamentos emergentes, os métodos descritos trazem ferramentas sólidas para alcançar esse objetivo.

Palavras-chave: árvores comportamentais; utilidade; reescrita de grafos; jogos eletrônicos.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), andregoncalves@unesc.net

²Orientador. Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), andre.ruaro@unesc.net

ABSTRACT: Since its conception, the people responsible for the development of video games have researched ways to introduce non-player characters capable of showing (or faking) intelligent and dynamic behaviour. Of the many ways of implementing those, the most common uses some kind of state management, with the finite state machine being the most popular example. Behavioral trees, although a superior alternative, also has its own limitations. The purpose of this research is to implement an evolution of the behavioural tree using needs and emotional states as a representation of utility in a way that allows for the reorganization of those trees, while overcoming the difficulties brought up by Colledanchise and Ögren's previous works on the subject regarding the usage of utility. The method employed will be the development of a playable prototype capable of showing the techniques that were used, its strengths and limitations, allowing for the creation of video games with characters capable of showing intelligent behaviour, even without a clearly designed relationship between agents, bringing a dynamic and fun experience. The results shown support the viability of the described techniques, while clearly showing its strengths and limitations. Although this research brings up limitations and serious considerations for the introduction of the techniques described, we have demonstrated that, in projects where the focus lies in intelligent behaviour, emergent behaviour and the relationship between entities, the methods described bring solid tools to reach this goal.

Keywords: behaviour trees; utility theory; graph rewriting; video games.

1 INTRODUÇÃO

No ramo da computação, as árvores comportamentais surgiram como uma estrutura alternativa de controle de estados modular e robusta aos autômatos finitos, otimizando o trabalho necessário para acompanhar o número exponencial de transições entre estados inerentes aos autômatos (Colledanchise; Ögren, 2016). Após atingir um nível de estabilidade maior, a técnica deixou de ser limitada apenas ao ramo dos jogos eletrônicos e passou a ser utilizada em diversas outras áreas, a robótica se mostrando particularmente notável (Colledanchise; Ögren, 2016).

As árvores comportamentais se destacam pelo nível de praticidade e escalabilidade em sua estrutura de comportamentos, permitindo tomada de decisão de forma hierárquica e separando a lógica de escolha do comportamento realizado, evitando ineficiências com gerenciamento de estado (Sekhavat, 2017).

Uma das fraquezas da técnica é a rigidez após o design, ou seja, por mais que sua implementação e alteração sejam relativamente simples, as regras que gerenciam o comportamento do agente não mudam em tempo de execução (Colledanchise; Ögren, 2022). Algumas soluções para esse problema foram propostas e implementadas, uma delas são as AC de Utilidade, que permitem a reorganização de determinados nós com base em um valor de utilidade esperado, porém esse tipo de implementação trás seus próprios desafios, o mais notável sendo a forma de determinar o nível de utilidade de cada decisão (Colledanchise; Ögren, 2022). Algumas outras soluções foram propostas e exploradas, como a parametrização das ACs (Shoulson et al, 2011), ACs Emocionais (Johansson; Dell'Acqua, 2012), entre outras.

Essa pesquisa traz como proposta uma exploração mais profunda das ACs emocionais, utilizando um ou mais valores emocionais esperados como abstração de utilidade e reestruturando a árvore comportamental para uma tomada de decisão que melhor supre as necessidades do agente, assim como calibrando os valores de alteração emocional esperada com base nas decisões anteriores, dessa forma gerando agentes capazes de tomar decisões coerentes, assim como a possibilidade de criar personagens não jogáveis mais dinâmicos na área dos jogos.

Uma pesquisa geral na área é necessária, apesar de com grande foco em diferentes implementações do conceito de árvore de comportamento, é necessário um destaque nas diferentes implementações no estudo da dinamização das ACs, incluindo métodos de utilidade, reorganização da árvore e possíveis expansões. A literatura abrange, em sua grande maioria, árvores comportamentais com emprego de utilidade, apesar de uma implementação diferente, é necessário reconhecer a influência dessa abordagem para esta pesquisa. Além das ACs, também é necessário uma pesquisa mais ampla, sendo preciso estudos de reorganização e reescrita de grafos, tanto em criação como tempo de execução, maneiras de codificar abstrações para estados emocionais, necessidades, relações e sentidos.

Essa pesquisa tem como objetivo demonstrar a aplicabilidade de estados emocionais como fator de utilidade em árvores comportamentais, assim como ferramenta eficiente de criação de memórias para criação de agentes inteligentes e complexos em jogos eletrônicos. Para isso, é necessária uma compreensão e implementação do conceito de árvore comportamental, reorganização da mesma através de abstrações de utilidade, assim como agentes autônomos em um ambiente apto à demonstração de

raciocínio.

Atualmente, as ACs podem aparentar rígidas e previsíveis, possivelmente diminuindo o nível de imersão dos jogadores (Johansson; Dell'Acqua, 2012). Baseado nos estudos, espera-se que a utilização de simuladores de emoção podem trazer comportamentos mais variáveis e, contraintuitivamente, mais racionais, onde os agentes individuais conseguem reagir de forma muito mais coerente aos estímulos locais.

Expandindo nos trabalhos anteriores, pretende-se que existem possibilidades a serem exploradas no uso de emoções não só como determinante da forma de tomar uma decisão, como um determinante da decisão em si, reorganizando os nós da AC para suprir as necessidades do agente.

Também espera-se que, ao utilizar uma implementação de estados emocionais como determinante de decisões, conseguimos implementar um método de criação de memórias eficiente e realista, realizando uma análise do estado emocional depois de determinado estímulo, considerando o ambiente em volta, criando uma relação de causa e efeito capaz de afetar futuras decisões envolvendo ambientes e estímulos semelhantes.

1.1 ÁRVORES DE COMPORTAMENTO

Conforme a necessidade de agentes autônomos mais inteligentes foi surgindo, formas mais eficientes de tomada de decisões se tornaram necessárias.

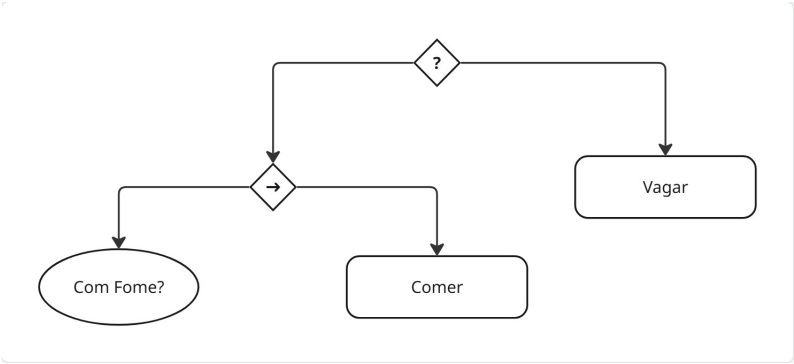
Árvores de comportamento surgem como uma estrutura reativa e modular de gerenciamento de tarefas, decisões e estados complexos. Inicialmente desenvolvidos para a área dos jogos eletrônicos, que comumente utilizavam autômatos finitos para esse tipo de tarefa, as ACs passaram a atrair atenção no meio acadêmico devido a sua modularidade, permitindo maior reusabilidade e expansibilidade (Colledanchise; Ögren, 2022).

Podemos construir uma AC utilizando uma série de componentes, utilizando um grafo direcionado com estrutura de árvore, utilizando nós de **Controle**, **Ação** e **Condição**, cada um retornando um valor de **Sucesso**, **Falha**, ou **Em Execução**. Como unidade de tempo atômica, são utilizados os **Ticks**, onde cada verificação é feita no intervalo de tempo ditado pelo mesmo. Uma demonstração da árvore com seus componentes pode ser observada na Figura 1 e Figura 2.

Em uma execução padrão, à cada **Tick**, o algoritmo verifica o nó atual, retornando o valor atual de retorno de seus nós filhos, caso os nós

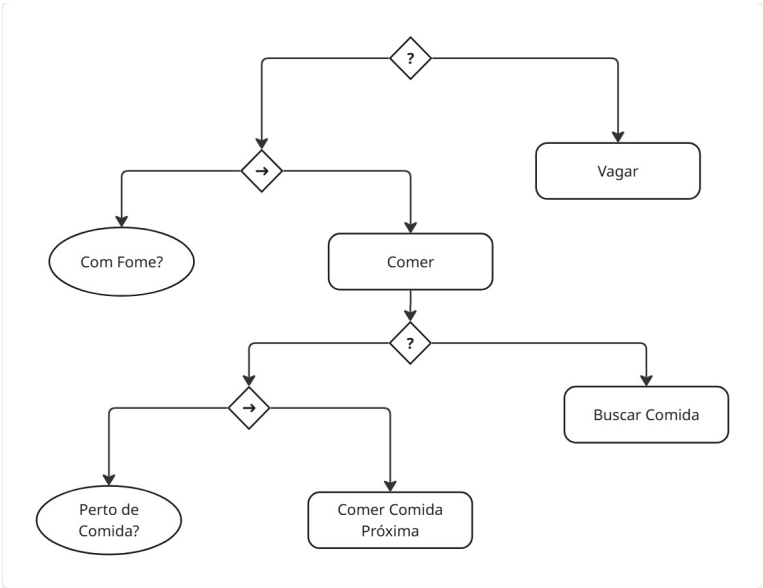
filhos não tenham um valor de retorno, continua a busca até um nó retornar um valor válido, em um nó de **Ação**, retornando se a ação em questão foi executada com sucesso, ou se ainda está em execução. Em um nó de **Condição**, o valor será determinado apenas se a condição for verdadeira ou não. Em um nó de **Controle**, o valor de retorno depende do controle definido, retornando **Sucesso** caso todos, ou apenas um nó filho retorne **Sucesso**, equivalente aos operadores lógicos **Ou** e **E**. A Figura 3 descreve um exemplo de árvore completa, mapeando um agente simulado.

Figura 1 - Representação de uma árvore comportamental básica, onde, caso a condição *com fome* retorne sucesso, a ação *comer* será tomada, caso contrário, o agente irá *Vagar*.



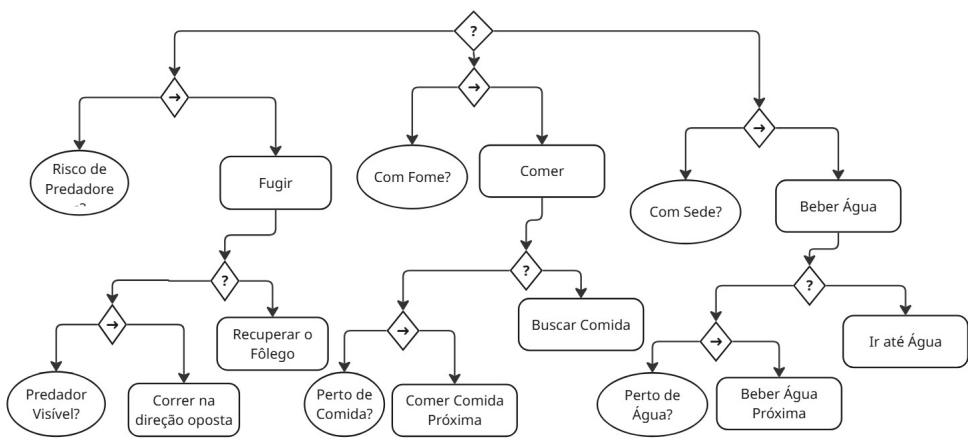
Fonte: Dos autores.

Figura 2 - Expansão do comportamento anterior, onde, caso a condição *Perto de Comida* retorne sucesso, realizará a ação *comer*, caso contrário, realizará a ação *Buscar Comida*.



Fonte: Dos autores.

Figura 3 - Árvore completa representativa de um pequeno herbívoro simulado, note a sequência fixa de ramos.



Fonte: Dos autores.

1.2 ÁRVORE COMPORTAMENTAL DE UTILIDADE

Pela natureza das ACs, um problema recorrente de rigidez se destaca, onde existem limitações em relação à mudança na ordem de prioridade dos nós da árvore, limitando possibilidade de um comportamento otimizado em um leque maior de situações.

Uma proposta para solucionar essa limitação é a utilização de uma medição de utilidade, dando prioridade maior às ações com um valor de utilidade maior. Essa abordagem é particularmente útil utilizando um nó de controle **Ou**, onde existem múltiplas ações com um retorno esperado de sucesso, mas uma das abordagens teria um resultado mais favorável. Um problema escondido nessa abordagem é a forma de medir a utilidade, que adiciona complexidade inusitada no sistema (Colledanchise; Ögren, 2022).

1.2.1 Modificação de Árvore Comportamental

Uma alternativa às ACs de utilidade, que tomam uma decisão com base em um escopo limitado e apenas determinando a decisão a ser tomada com base na utilidade esperada, é a modificação da AC, adicionando, removendo ou reordenando nós conforme a situação ou variação entre agentes, permitindo que mudem suas prioridades em tempo de execução.

1.2.2 Árvore Comportamental emocional

Uma ferramenta para determinar utilidade explorada em outros trabalhos é a utilização de redes de comportamentos emocionais, como explorado por (Johansson; Dell'Acqua, 2012) e outros autores. Porém, explorações atuais se limitam em apenas fazer uma comparação entre as abordagens, ou apenas como um determinante interno semelhante às ACs de utilidade, adicionando uma complexidade ainda maior, sem benefícios significantes comparados à uma escolha aleatória. Revisitar esse tópico com uma nova abordagem pode trazer resultados mais dinâmicos para os jogos eletrônicos, assim como uma tomada de decisão adaptativa para ambientes da robótica.

1.3 DEFINIÇÃO DE ESTADO EMOCIONAL

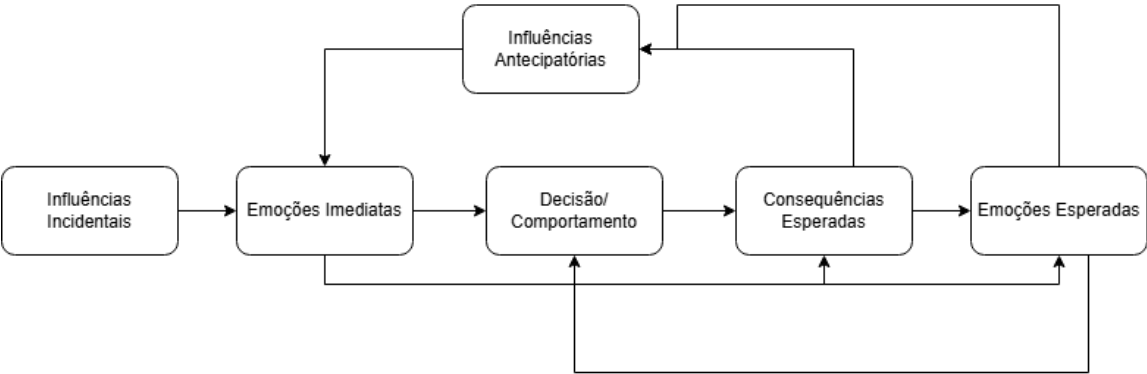
A principal dificuldade desse trabalho se encontra na forma de definir, assim como codificar emoção e estados emocionais. Apesar do uso da palavra "emoção" para um agente digital, pode ser substituído o termo **medo** por **foco em auto-preservação** ou equivalente, podendo inferir que qualquer evento que danifique o agente de alguma forma afete o parâmetro **medo**. Esse parâmetro, junto com uma série de outros, definiriam se um rumo de ação deve ser tomado, mesmo que exista o risco de danos.

Um desafio surge nessa abordagem, sendo o número de emoções que um agente deve possuir, de forma que um número alto demais adicione complexidade e reduza a autoridade de cada uma e um número

muito baixo torne o agente incapaz de adaptação. Além disso, é necessário definir um delta para cada estímulo, como decidir o nível de correlação entre danos infligidos e a modificação do parâmetro relacionado? Uma possível solução é uma projeção cuidadosa por parte do desenvolvedor, que possui limitações práticas, alternativamente, é possível utilizar algoritmos inspirados em computação evolutiva (Pozo et al. 2012) para geração de múltiplos tipos de agentes viáveis. Nossa abordagem consiste em diferentes graus de ambas.

Para a reorganização da árvore comportamental, foi usado uma unidade de tempo maior que o **Tick** padrão, chamada a partir de agora de **Tick Emocional** ou **eTick**, de forma que, à cada N ticks, se passará um eTick. À cada eTick, será feita uma chamada para reorganização dos nós da árvore, de forma a reduzir emoções negativas e aumentar emoções positivas da forma mais eficiente possível. A principal forma de determinar o curso de ação mais eficiente consiste em analisar experiências e ações anteriores e definir o delta das emoções resultantes desse rumo. Para isso, foi tomada forte inspiração dos trabalhos de (Loewenstein e Lerner, 2003) seu modelo de emoções, demonstrado na Figura 4 ,tomando decisões com base nas emoções atuais e emoções esperadas depois de determinada ação. Após a ação ser concluída, foi verificado as mudanças de emoção de determinada ação e é utilizada essa memória para futuras decisões.

Figura 4 - Modelo de emoções e sua relação com decisões futuras.



Fonte: Loewenstein e Lerner (2003).

Para a criação de memórias, foi decidido utilizar um formato de dado simples, de forma a suportar um grande número de agentes simultâneos na nossa simulação. Com esse objetivo, memórias constituem de alguns elementos, sendo eles: **Tempo**, **Local**, **Delta Emocional**, **Última Ação** e **Entidades Próximas**.

Como exemplo, pode-se imaginar um rato com fome, devido à experiências anteriores, o rato sabe que se for à cozinha da casa mais próxima, saciará sua fome e, devido à qualidade da comida, aumentará sua felicidade. Sem seu conhecimento, o morador da casa adotou um gato, com seus próprios comportamentos. Nunca tendo encontrado um gato antes, o rato consegue se alimentar, mas dessa vez menos que o normal e acaba com ferimentos leves. Após esse evento, uma nova memória é criada no agente, atualizando sua perspectiva do ambiente, sabendo que a fome saciada agora é menor, e existe uma nova ameaça causadora de medo. Ao longo de diversas memórias, o rato saberá que qualquer ação tomada que pode envolver um gato deve ter uma prioridade muito menor, tomada apenas em casos extremos.

No âmbito dos jogos, é desejável que as espécies presentes sejam divertidas para o jogador, apesar de desafiante medir o nível de entretenimento de um jogador humano, parte-se do princípio que a diversidade de cenários emergentes seja uma influência positiva para o mesmo. Seguindo na interação jogador-ambiente, foi observado que um nível de erro por parte dos agentes seja preferível, influenciando algumas das decisões tomadas durante o processo de *design*.

2 TRABALHOS CORRELATOS

Esse trabalho teve inspiração direta de trabalhos da área, assim como influência de diversos artigos semelhantes, em busca de aprofundar o leque de ferramentas disponíveis.

Behavior Trees in Robotics and AI: An Introduction, publicado por Michele Colledanchise e Petter Ogren, consiste na principal base deste trabalho, expondo grande parte do conhecimento prévio necessário para o início da pesquisa. O trabalho introduz e explica o conceito das Árvores de comportamento, assim como sua história, vantagens, limitações e expansões (Colledanchise; Ögren, 2022). Esse livro, também inclui os trabalhos: *How Behavior Trees Modularize Robustness and Safety in Hybrid Systems* (2014), *Increasing Modularity of UAV Control Systems using Computer Game Behavior Trees* (2012) e *How Behavior Trees Generalize the Telemorphic Paradigm and And-Or-Trees* (2016).

Comparing Behavior Trees and Emotional Behavior Networks for NPCs, publicado por Anja Johansson e Pierangelo Dell'Acqua, o trabalho faz uma exploração e compara as árvores de comportamento e redes de comportamento emocional, fazendo uma explicação de ambas, descre-

vendo seu comportamento, assim como referenciando ambientes no qual foram utilizadas. Também é feita uma comparação qualitativa entre ambas, exemplificando a implementação para diversas situações e demonstrando o comportamento em relação ao planejamento, previsibilidade, entre outras características (Johansson, Anja; Dell'Acqua, Pierangelo, 2012).

Emotional Behavior Trees, também por Anja Johansson e Pierangelo Dell'Acqua, o trabalho explora a utilização de emoção para tomar decisões em uma árvore de comportamento, incluindo padrões onde apenas algumas decisões são afetadas por emoções, assim como definir a ação a ser tomada com base nas emoções atuais junto com o tempo de execução e risco esperado. Pode-se afirmar que esse trabalho é o mais próximo em tópico explorado, porém observamos possíveis limitações na abordagem escolhida (Johansson, Anja; Dell'Acqua, Pierangelo, 2012).

Computação Evolutiva, por Aurora Pozo, Andrea de Fatima Cavalheiro, Celso Ishida, Eduardo Spinoso e Ernesto Malta Rodrigues, o trabalho explora e se aprofunda em diversas técnicas e abordagens utilizando algoritmos genéticos, de forma que, com um conjunto inicial de agentes, uma tarefa com resultado quantificável, meio de seleção e mutação, consegue gerar agentes cujo comportamento se aproxima ao ótimo à cada geração (Pozo, Aurora et al, 2012).

3 MATERIAIS E MÉTODOS

3.1 TRABALHO PROPOSTO

O desenvolvimento da pesquisa foi feito com base em uma simulação jogável utilizando a *engine* Godot, devido a sua demanda baixa de recursos, baixo custo e filosofia naturalmente compatível com o uso de grafos de forma visual.

Para criação dos agentes, foi necessário criar todas as emoções disponíveis, juntos com suas interações base com diversos estímulos. Também existe a possibilidade de, em algumas espécies, a criação dinâmica de reações a certos estímulos (uma espécie diferente de rato pode ficar feliz, ao ser atacado, por exemplo). Para uma maior variação, é determinado atributos físicos, além de comportamentais, esses são mais simples, apenas para interação com o mundo. Também foi criado espaço de possibilidade de uma emoção ou rumo de ação não ser gerado em determinada espécie (Um rato que não sente medo).

Para que os estímulos existentes, assim como as emoções de cada espécie tenham qualquer utilidade, formas de interagir com o mundo

foram implementadas, incluindo outros agentes. Além disso, foi necessária a implementação de necessidades para os mesmos, de forma a obrigar os agentes a interagir com o mundo.

Na simulação descrita, o ambiente também precisou ser implementado, podendo variar em diversos fatores de forma a estimular diferentes tipos de espécies. Na exploração feita, o ambiente consiste em poucas peças distintas, mas que demonstram a capacidade dos agentes.

Após todas as peças tenham sido corretamente criadas e implementadas, o algoritmo de geração de memórias pode ser introduzido, podendo finalmente adicionar dinamicidade para o comportamento dos agentes. Também foi necessário adicionar novas ações para dar suporte à todos os passos anteriores.

Uma série de maneiras de visualizar a árvore de comportamento, assim como as necessidades, emoções e memória dos agentes precisou ser implementada.

Por último, foi feita uma observação avaliativa dos diversos elementos propostos na pesquisa, determinando se o comportamento dos agentes trouxe as qualidades referidas, funcionou corretamente e as consequências de sua implementação tornaram um ambiente jogável mais dinâmico.

3.2 ANÁLISE E SELEÇÃO DE FERRAMENTAS

Dado o foco em jogos eletrônicos da pesquisa, uma das decisões tomadas é a escolha de ferramentas para a prototipagem do mesmo, isso inclui a lógica do jogo em si, assim como recursos de arte, áudio, etc. A escolha de *game engine* se encontrou principal em meio as outras decisões necessárias, considerando a sua influência no tempo de desenvolvimento, tecnologias disponíveis, custo monetário e linguagem de programação (assim como, até certo nível, afetar a performance do protótipo final).

3.3 PROTOTIPAÇÃO DE AGENTES

Devido ao número de peças simultâneas e a relação entre elas, a definição do primeiro passo à ser codificado é uma tarefa não trivial. Foi então definido a seguinte ordem de dependências:

Primeiro foi feita a prototipação de um algoritmo de valor emocional e de necessidade, podendo ser representado como um valor numérico com uma faixa máxima e mínima, mas sem necessariamente um comportamento associado. Esses valores serviram como *alavancas* para os diversos

comportamentos futuros.

Após os valores emocionais, um conjunto de ações atômicas foi implementado, isso inclui atividades simples como *comer*, *mover* e *descansar*. Essas ações foram mapeadas à diversas emoções e necessidades, afetando positivamente uma emoção positiva e suprimindo uma necessidade física.

As ações atômicas, e seus parâmetros emocionais não são particularmente úteis sem um ambiente que teste esses comportamentos, para isso, além da criação de um ambiente propício, uma implementação simples de gasto calórico é necessária. Nessa implementação, existe uma queima passiva de calorias, que podem ser recuperadas comendo. Graças à fácil expansão das ACs, esses comportamentos podem ser ampliados para um crescimento em ambientes mais desafiadores.

Com os passos previamente citados, os agentes, apesar de capazes de realizar ações de forma consistente, não possuem dinamicidade, não são capazes de associar as necessidades com as ações, nem são capazes de priorizar comportamentos com base em risco, prioridades, etc. Para isso foi aplicada a técnica de resultado esperado, onde o agente em questão faz uma estimativa da mudança emocional de determinado rumo. Os valores de estimativa inicial são definidos manualmente e não representam completamente o ambiente que se encontram, comparável aos instintos de um animal recém nascido.

Por último, é prototipado a criação e reforço de memórias, permitindo o agente atualizar os valores esperados de determinada ação, com base no comportamento escolhido e seu contexto, de forma a entender, por exemplo, onde buscar comida de forma mais segura, que predadores evitar, etc. Para o auxílio ao desenvolvedor, ferramentas de *debug* foram criadas, assim como refinadas para visualização da tomada de decisão dos agentes, agindo como mais uma forma de transmitir o conhecimento sobre as árvores de comportamento.

3.4 OBSERVAÇÕES QUALITATIVAS

Com um protótipo feito , com as escolhas de design refinadas à um nível aceitável, é necessário testar se as decisões tomadas e as técnicas aplicadas realmente realizam os objetivos propostos nesse trabalho, definindo se os comportamentos são complexos e realmente demonstram racionalidade, se essas características tornam a experiência mais dinâmica. Em casos de jogos eletrônicos, tornam a experiencia mais divertida, assim

como em casos de robôs em um ambiente de trabalho, essas técnicas produzem o resultado desejado.

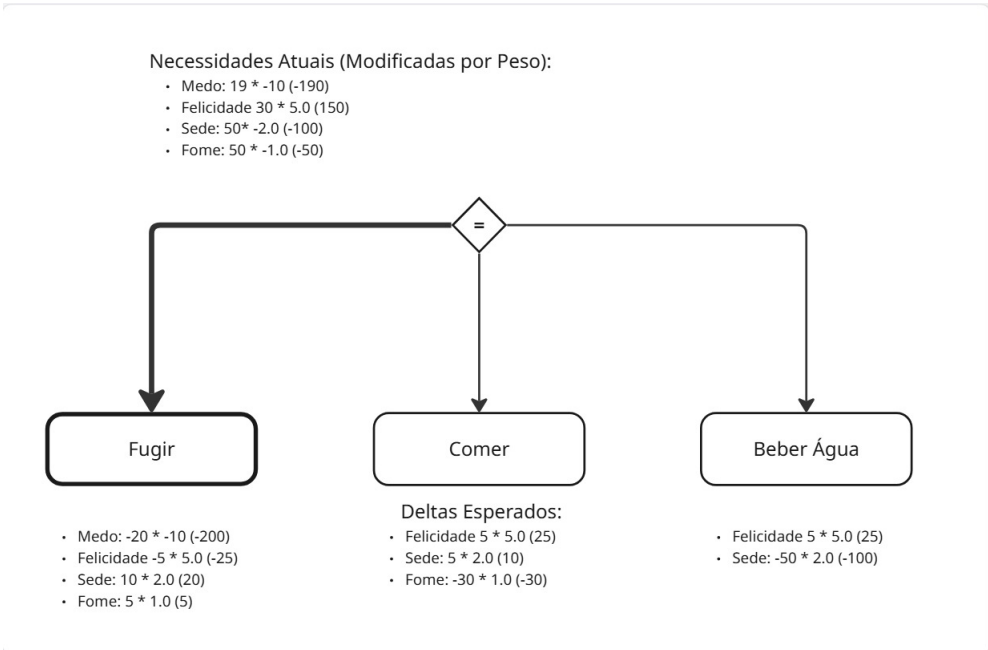
4 RESULTADOS E DISCUSSÃO

A utilização de emoção e necessidade como um fator análogo à utilidade para a reestruturação de ACs demonstrou capacidades esperadas no âmbito dos jogos eletrônicos, a reorganização de nós da AC com base nas necessidades atuais trás uma habilidade de tomada de decisões satisfatória em comparação com ACs não emocionais, sendo possível notar um comportamento mais dinâmico, imprevisível, porém coerente nos olhos do jogador.

A implementação do estado emocional consistiu na criação de um tipo de dado apelidado **Need** (Necessidade), com uma série de valores numéricos para manipulação e interação com a árvore. Dentre esses valores, existem: **Máximo**; **Mínimo**; **Padrão**, representando o valor inicial; **Atual**, apenas utilizado em *runtime*; **Delta por tick**, representando uma modificação passiva, particularmente útil para necessidades como fome; **Delta direcionado**, representando um segundo tipo de modificação passiva, porém em direção à um valor específico, geralmente o valor padrão; **Modificador de mudança**, afetando em porcentagem o quanto o valor muda, em comparação com outros membros da mesma espécie; **Necessidades afetadas**, representando outras emoções que recebem alterações em cascata, por exemplo, o parâmetro fome pode, ao aumentar, diminuir proporcionalmente o parâmetro felicidade; Por último, foi introduzido um modificador de **Peso**, referenciando o quão prioritário o valor dessa propriedade é. Por exemplo, pode ser definido que o parâmetro sede tem uma prioridade maior à fome. É possível também utilizar um valor negativo, demonstrando prioridade para *reduzir* um valor.

Para permitir que as necessidades e emoções influenciem a tomada de decisão, um novo tipo de nó de controle foi criado na árvore comportamental, exemplificado na Figura 5. Apelidado de **nó emocional**, esse nó não possui uma equivalência com uma estrutura de controle padrão como **E** ou **Ou**. Ao invés disso, seleciona um dos ramos à seguir com base nas alterações emocionais esperadas em cada caminho, modificado pela prioridade emocional atual. Após a execução, o nó é responsável por atualizar o valor esperado do ramo, conforme o resultado obtido por sua execução.

Figura 5 - Reescrita da árvore descrita na Figura 3, com utilização do novo nó emocional (=). Note a falta de condicionais como **Com fome?**.

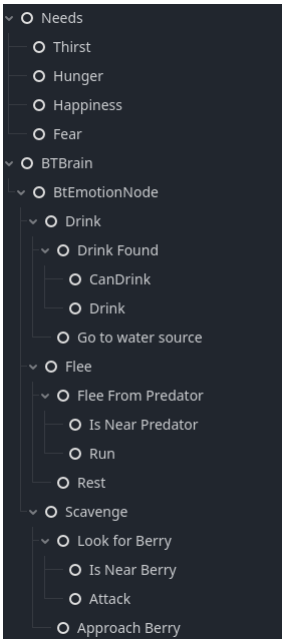


Fonte: Dos autores.

Uma das consequências positivas dessa implementação é a facilidade da formação de *personalidades* entre agentes da mesma espécie, é possível, por exemplo, criar um arquétipo de personalidade *mal-humorado* associando um modificador de valores tão baixos quando cinco por cento no aumento do parâmetro **medo** ou **raiva** em resposta à estímulos comparados à outros membros da mesma espécie. Dado o número de parâmetros e a leque de variações numéricas possíveis, a variedade entre agentes da mesma espécie (ao mesmo tempo mantendo os comportamentos comuns) é extremamente bem vinda.

Uma limitação esperada das ACs como um todo, é a necessidade de desenvolver todas as ações atômicas realizáveis pelos agentes, esse trabalho não é exceção. Enquanto o agente pode tomar a decisão **comer** (ao invés de qualquer outra ação possível), A ação em si deverá ser programada individualmente (dependendo do nível de complexidade da ação atômica, o algoritmo permite a utilização de uma AC dedicada à essa ação). Em contrapartida, se torna mais um ângulo para uma maior variedade de espécies diferentes, uma árvore comportamental de uma dessas espécies é descrita na Figura 6.

Figura 6 - Representação da árvore descrita na Figura 3, acompanhada de necessidades na *engine* GODOT.



Fonte: Dos autores.

Assim como a variação entre membros da mesma espécie, são observáveis comportamentos inteiramente diferentes entre agentes de diferentes espécies pela existência (ou não) de algum parâmetro, implementação de suas ações atômicas (um carnívoro realiza a ação **comer** de forma diferente de um herbívoro) ou até mesmo a relação entre suas características físicas com suas memórias (uma criatura mais frágil *provavelmente* não terá boas memórias envolvendo confrontos físicos e escolherá uma abordagem diferente).

A geração de memórias, apesar de ser uma ferramenta potente para a tomada de decisões futuras, trouxe diversos desafios que dificultaram a implementação desejada. Dentre essas limitações, as mais influentes foram a complexidade de implementação e o custo computacional. Apesar de ser desejado revisitar essa implementação no futuro, foi decidido a utilização de uma abordagem consistindo apenas de **Resultado Esperado** e **Resultado Obtido**.

Uma consideração importante em relação ao uso das técnicas descritas é o trabalho por parte dos desenvolvedores e **designers** dos agentes individuais, de forma à permitir cada agente à ter um leque grande de ações atômicas. Também é aconselhado um nível de aleatoriedade entre parâmetros de membros da mesma espécie. Uma implementação que

peça nesses pontos prejudica a experiência, considerando que um agente inepto e um agente que *aparenta* inepto são, essencialmente, a mesma coisa nesse contexto.

5 CONCLUSÃO

Neste trabalho foi proposto uma implementação capaz de utilizar parâmetros de utilidade para a reorganização de árvores comportamentais, trazendo soluções viáveis aos desafios propostos por outros autores. A utilização levantada de estados emocionais e necessidades, agregada com o algoritmo de geração de memórias, nos permite trazer comportamentos extremamente dinâmicos e imprevisíveis, enquanto coerentes e lúdicos no âmbito dos jogos eletrônicos.

A utilização do novo nó emocional, assim como sua relação com os parâmetros providenciados, cumpre o papel proposto de modificar a árvore comportamental em tempo de execução. A simulação providenciada demonstra esse comportamento de forma clara, demonstrando adaptação no comportamento dos agentes em resposta à mudanças no ambiente.

Os métodos descritos, enquanto realizam seu propósito de forma satisfatória, demandam alta cautela na decisão de sua implementação, visto a alta complexidade envolvida, assim como custo computacional significativo em sua execução. O sistema demonstra melhor suas forças em ambientes com um número menor de agentes simultâneos, com um foco maior nos seus comportamentos e relações, apesar disso, sugestões para um número maior de agentes é providenciada.

Uma limitação à ser considerada é o número de agentes processados em simultâneo, considerando a demanda computacional de cada agente, cada um com sua própria AC executando em paralelo, assim como os custos de RAM necessários para o armazenamento de memórias. Cautela é aconselhada ao tentar implementar um grande número de agentes, o sistema mostra suas forças em cenários com um número menor de agentes mais inteligentes, do que o oposto. Além de otimizações gerais, o aumento de tempo entre **Ticks Emocionais**, assim como uma redução da capacidade individual de armazenamento de memórias reduzirá a demanda computacional (sacrificando dinamicidade). Essas limitações são mais prevalentes em aplicações em tempo real, um jogo de turnos, por exemplo é capaz de suportar um número muito maior de agentes simultâneos.

É desejado, no futuro, uma nova exploração do método de armazenamento de memórias que, apesar de capaz de cumprir sua função

de forma satisfatória, não fez jus ao que acredita-se ser seu potencial completo. Também foi identificados os autômatos finitos não determinísticos como um caminho viável para trabalhos futuros.

Com base nos resultados obtidos, a abordagem proposta se demonstra um forte candidato em aplicações onde o comportamento individual de cada agente é o foco da experiência. Também são levantados pontos à serem trabalhados em estudos futuros.

REFERÊNCIAS

ÖGREN, Petter; COLLEDANCHISE, Michele. **Behavior Trees in Robotics and AI: An Introduction**. 1. ed. [S.l.]: CRC Press, 2018. 206p.

SEKHAVAT, Yoonas A. **Behavior Trees for Computer Games**. International Journal on Artificial Intelligence Tools, v. 26, n. 02, p. 1730001, 2017. Disponível em: <<https://www.worldscientific.com/doi/abs/10.1142/S021821301730001>>. Acesso em: 13 out. 2024.

ROIJERS, Diederik Marijn; VAMPLEW, Peter; WHITESON, Shimon; et al. **A Survey of Multi-Objective Sequential Decision-Making**. 2014. Disponível em: <<https://arxiv.org/abs/1402.0590>>. Acesso em: 3 nov. 2024.

JOHANSSON, Anja; DELL'ACQUA, Pierangelo. **Comparing behavior trees and emotional behavior networks for NPCs**. In: 2012 17th International Conference on Computer Games (CGAMES). [s.l.: s.n.], 2012, p. 253–260. Disponível em: <<https://ieeexplore.ieee.org/document/6314584>>. Acesso em: 3 nov. 2024.

SHOULSON, Alexander; GARCIA, Francisco M.; JONES, Matthew; et al. **Parameterizing Behavior Trees**. In: **ALLBECK**, Jan M.; FALOUTSOS, Petros (Orgs.). Motion in Games. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, v. 7060, p. 144–155. Disponível em: <http://link.springer.com/10.1007/978-3-642-25090-3_13>. Acesso em: 3 nov. 2024.

COLLEDANCHISE, Michele; ÖGREN, Petter. **How Behavior Trees modularize robustness and safety in hybrid systems**. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. [s.l.: s.n.], 2014, p. 1482–1488. Disponível em: <<https://ieeexplore.ieee.org/document/6942752>>. Acesso em: 3 nov. 2024.

COLLEDANCHISE, Michele. **Behavior Trees in Robotics**. 2017. Disponível em: <<https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-202926>>. Acesso em: 3 nov. 2024.

COLLEDANCHISE, Michele; PARASURAMAN, Ramviyas; ÖGREN, Petter. **Learning of Behavior Trees for Autonomous Agents**. IEEE Transactions on Games, v. 11, n. 2, p. 183–189, 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8319483>>. Acesso em: 3 nov. 2024.

COLLEDANCHISE, Michele; ALMEIDA, Diogo; ÖGREN, Petter. **Towards Blended Reactive Planning and Acting using Behavior Trees**. In: 2019 International Conference on Robotics and Automation (ICRA). [s.l.: s.n.], 2019, p. 8839–8845. Disponível em: <<http://arxiv.org/abs/1611.00230>>. Acesso em: 3 nov. 2024.

COLLEDANCHISE, Michele; ÖGREN, Petter. **How Behavior Trees Generalize the Teleo-Reactive Paradigm and And-Or-Trees**. In: [s.l.: s.n.], 2016. Disponível em: <<https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-194179>>. Acesso em: 3 nov. 2024.

GDC 2005 **Proceeding: Handling Complexity in the Halo 2 AI**. Disponível em: <<https://www.gamedeveloper.com/programming/gdc-2005-proceeding-handling-complexity-in-the-i-halo-2-i-ai>>. Acesso em: 3 nov. 2024.

JOHANSSON, Anja; DELL'ACQUA, Pierangelo. **Emotional behavior trees**. In: 2012 IEEE Conference on Computational Intelligence and Games (CIG). [s.l.: s.n.], 2012, p. 355–362. Disponível em: <<https://ieeexplore.ieee.org/document/6374177>>. Acesso em: 3 nov. 2024.

LOEWENSTEIN, George; LERNER, Jennifer. **THE ROLE OF AFFECT IN DECISION MAKING**. In: 2012 IEEE Conference In R.J. Davidson, K.R. Scherer & H H. Goldsmith (2003), Handbook of Affective Sciences Oxford University Press. 2003, p. 355–362. Disponível em: <https://www.contrib.andrew.cmu.edu/~gl20/GeorgeLoewenstein/Papers_files/pdf/RoleofEffectEmotion.pdf>. Acesso em: 6 jun. 2025.

POZO, Aurora et al. **COMPUTAÇÃO EVOLUTIVA**. In: Grupo de Pesquisas em Computação Evolutiva. [s.l.: s.n.], 2012. Disponível em: <<https://www.inf.ufpr.br/aurora/disciplinas/datamining/Ceapostila.pdf>>. Acesso em: 6 jun. 2025.