

# O Behaviour-Oriented Design e seu potencial para agentes em jogos digitais

André A. N. Alves  
Universidade Estadual de Campinas

Vinicius D. O. Gardelli  
Universidade Estadual de Campinas

Bruno A. Melo  
Universidade Estadual de Campinas

## Resumo

O ramo da inteligência artificial (IA) possui diversas aplicações em jogos. Nem sempre uma IA avançada é necessária, muitos jogos podem contornar a situação usando condicionais simples, ou usando um pouco de inovação no design, uma IA avançada pode ser obtida sem a necessidade de algo mais elaborado.

Esse artigo tem como objetivo tratar os casos onde uma IA avançada é necessária, propondo o uso da metodologia Behaviour Oriented Design (BOD), proposta por Joanna J. Bryson em sua tese de doutorado [Bryson 2001] e revista em um artigo publicado posteriormente [Bryson 2003], para modelagem de agentes, proporcionando uma maneira simples mas muito poderosa de implementar uma IA elaborada para certos jogos. Para isso serão apresentados neste artigo alguns dos mecanismos da implementação que foram trabalhados.

**Keywords::** Agentes, Projeto de Agentes, Agentes Comportamentais, Projeto Orientado a Comportamentos, BOD, Behavior-Oriented Design

## Author's Contact:

alves.andreaugusto@gmail.com  
vinicius.gardelli@gmail.com  
brunoamelo@gmail.com

## 1 Introdução

Jogos são criados para entretenimento. Nem sempre uma IA avançada é necessária, mas em certos casos a ausência de algo mais elaborado torna-se um problema para cumprir o objetivo final. O tempo para produção de um jogo pode ser determinante e a falta de conhecimentos específicos da equipe na área de IA pode inviabilizar muitos recursos que seriam interessantes para um jogo.

Propõe-se então o uso da metodologia do BOD, que aborda a construção de agentes dentro de IA. Tal metodologia propõe um modo de desenvolvimento de sistemas de agentes de forma simples e organizada, permitindo uma implementação que pode atender às necessidades de jogos mais simples até os mais complexos. Ao mesmo tempo, o BOD não exige conhecimentos avançados de IA, podendo ser utilizado por programadores com pouca ou nenhuma experiência na área.

## 2 Trabalhos Relacionados

Além dos próprios trabalhos de Bryson [Bryson 2001; Bryson 2003] que utilizam o BOD, um dos grandes motivadores para os trabalhos realizados foi o [Grey and Bryson 2009], que propõe uma implementação de agentes sociais (Believable Social Agents - BSA), capazes de interagir dentro de comunidades autônomas criadas no jogo e proporem quests (ou aventuras) geradas em tempo de execução pelos agentes, baseados nas experiências que viveram no decorrer do jogo e paralelamente às ações executadas pelo jogador. Uma vez que essa implementação seja atingida, abre-se espaço para uma melhoria significativa no relacionamento entre jogadores e agentes (doravante denominados simplesmente de NPCs, de acordo com definição posterior neste texto), proporcionando uma maior imersão e novas possibilidades de desenvolvimento da história de pano de fundo para o jogo, que agora poderia ser vivida e desenvolvida pelos próprios agentes sociais e apresentada ao jogador.

X SBGames - Salvador - BA, November 7th - 9th, 2011

## 3 Definições Iniciais

Para melhor explicar a metodologia e os modelos testados desenvolvidos, faremos primeiro uma rápida introdução sobre agentes e algumas definições básicas de termos utilizados ao longo do artigo, como visto em [Russel and Norvig 2003].

Um agente é uma entidade que, dado um ambiente em que se encontra, possui sensores que lhe permitem extrair informações desse ambiente, atuadores que lhe permitem interagir com esse ambiente e mecanismos para o processamento das informações adquiridas. Por exemplo, seres humanos são agentes no ambiente em que vivemos.

A partir dessa definição é possível inferir que a utilização de agentes pode ser ideal para alguns jogos, uma vez que Non-Player Characters (NPCs), personagens que sejam comandados e/ou controlados pelo computador, podem ser modelados dessa forma. Os próprios inimigos controlados pelo computador (popularmente conhecidos como "bots"), em jogos de tiro em primeira pessoa (ou FPS) são considerados agentes.

Existem diversos tipos de agentes e as definições completas de todos fogem ao escopo do artigo. É suficiente conhecer a definição simples de agentes reativos e planejadores, como apresentadas abaixo, para que o leitor entenda os resultados obtidos e a metodologia proposta. O Behaviour-Oriented Design (BOD) propõe a criação dos ditos agentes reativos, que são, numa visão simplista, todos aqueles que reagem ao ambiente dadas certas condições. Por exemplo, um agente possui uma condição "Encontrei Inimigo" que desencadeia a reação "Atacar Inimigo".

Tais agentes estão entre os mais simples existentes. Assim, o BOD vai um pouco além propondo viabilizar aos agentes a capacidade de planejamento. Agentes planejadores são mais complexos e demandam muito mais trabalho que uma implementação puramente reativa. Um agente planejador é capaz de guiar suas ações para que determinado plano seja cumprido, ou seja, é capaz de selecionar quais ações irá cumprir, ao invés de simplesmente reagir ao ambiente sempre da mesma forma. Também é capaz de atingir um objetivo executando planos de ação, no qual conjuntos de ações para solucionar o problema são estabelecidos e seguidos.

Outra definição importante refere-se à capacidade de comunicação entre os agentes. Um sistema com N agentes interagindo entre si é dito *multi-agentes*. Porém, sem que haja troca de mensagens entre os agentes desse ambiente, não podemos considerá-lo como tal. Implementar os agentes de forma a possibilitar tais trocas de mensagens é imprescindível para que situações como as descritas como BSA, em [Grey and Bryson 2009] sejam possíveis, visto que são um pré-requisito das interações sociais ali simuladas.

## 4 Behavior-Oriented Design e os Jogos

Feitas tais definições, pode-se seguir para uma rápida apresentação da metodologia de projeto orientada a comportamentos, do inglês Behaviour-Oriented Design (BOD). Aqui serão apresentados apenas os pontos principais da metodologia, com suporte de alguns exemplos. Tal metodologia em sua completude é extremamente extensa e a leitura da bibliografia é fortemente recomendada caso o leitor tenha interesse em aplicar a metodologia em seus trabalhos.

O BOD propõe tratar os mecanismos utilizados pelo agente como objetos: sensores, atuadores, memória, aprendizado, comportamentos e quaisquer outros mecanismos que possam ser identificados serão todos modelados como objetos. Conta-se com a definição de conceitos chamados de "Como", "Quando" e "O quê fazer", definidos posteriormente no texto, que auxiliarão na criação dos agentes.

A metodologia é então, como pode-se notar, amplamente fundamentada no paradigma de programação orientado a objetos (POO).

Para exemplificar melhor esse ponto será utilizada também uma aplicação desenvolvida pelos autores, utilizando o BOD.

Utilizando o framework XNA 4.0 para C#, os autores desenvolveram um jogo simples onde círculos de diversas cores são capazes de se mover pela tela, tratados como personagens em vista superior, e disparar pequenos projéteis uns nos outros. Ao jogador é permitido controlar um desses círculos usando o teclado para movimentação, mirar e disparar projéteis na direção dada pelo mouse.

Nesse caso, os sensores dos agentes foram representados e implementados através de círculos invisíveis com raios maiores que os agentes e concêntricos a eles, permitindo-lhes ter consciência de tudo que estivesse em seu interior, fossem projéteis ou outros agentes (incluindo o jogador). Assim, da mesma forma que o jogador possui um campo de visão dado pela câmera, os agentes poderiam “ver” ao seu redor.

Para os atuadores, os agentes receberam os mesmos controles que ao jogador: podem se mover em todas as direções num plano e disparar projéteis em qualquer direção que lhes seja conveniente.

Para processar as informações obtidas pelo sensores, foi desenvolvida uma memória de curta duração com o intuito de armazenar, durante períodos pré-determinados de tempo, os elementos detectados. Utilizando essa memória de curta duração, o agente fica independente de qualquer informação externa durante a tomada de decisões, e qualquer tipo de interação é executada através de seus atuadores, uma vez que as informações que serão usadas para tomar a decisão estão todas armazenadas nessa memória.

A tal elemento de memória foi atribuída uma duração muito curta (alguns milissegundos), assim, para informações que devem ser guardadas por um longo período, foi desenvolvida uma memória de longa duração, cujos dados não possuem tempo de expiração. Nos testes e simulações realizados, o potencial dessa memória foi pouco explorado, sendo utilizada apenas para guardar informações sobre aliados ou inimigos. Mas fica claro que essa memória, se bem utilizada, proporciona grandes possibilidades, por exemplo, de aprendizado para o agente.

A partir dessas estruturas, foram obtidos agentes bem modularizados, aos quais a adição de novas funcionalidades, como diferentes sensores e atuadores, foi simplificada. Um exemplo disso foi a adição de obstáculos em um momento avançado do desenvolvimento. Para tratá-los só bastou adicionar aos sensores a capacidade de interpretar sua existência. Um ponto importante a notar é que as trocas de informação entre componentes independe de sua implementação futura, assim, a troca de um sensor ou componente da memória por exemplo, não altera sua comunicação com outros pontos, evidenciando bem a modularidade descrita.

As estruturas descritas permitem ao agente “perceber” o ambiente, “interagir” com o mesmo e “armazenar informações”, mas ainda não dão nenhum tipo de tratamento para essas informações nem fornecem meios de permitir ao agente utilizar seus atuadores.

A partir dessas implementações básicas, foi iniciada a exploração dos mecanismos mais robustos do BOD, a fim de aproximar a implementação puramente reativa de agentes planejadores.

Um comportamento simples, como proposto pela metodologia, é definido como uma tripla composta por uma prioridade (que define qual comportamento deve ser priorizado caso duas condições diferentes tenham sido atendidas), uma condição a ser atendida para que um comportamento possa ser executado e uma ação, que nada mais é que a reação associada à condição. A partir dessa tripla, propõe-se estruturar a lista de comportamentos possíveis de serem realizados através de uma pilha, definida pelas prioridades dos comportamentos (os comportamentos de prioridade mais alta ficam no topo da pilha, e os testes de validade são feitos do topo para baixo, caso um deles seja válido, a verificação termina e o comportamento é selecionado.

A organização e documentação dos comportamentos acaba por tornar-se vital para o crescimento do projeto, uma vez que a criação

de uma pilha de comportamentos para determinado agente parte de uma descrição de alto nível do comportamento do agente. A partir dessas descrições, definem-se os comportamentos, com possibilidade de reutilização de comportamentos utilizados em outras pilhas, montando uma espécie de “biblioteca” de comportamentos pré-fabricados.

Dentro do ambiente criado para os testes já pode-se observar resultados muito bons com relação a essa abordagem, quando comparados à implementação ingênua inicial, através de métodos. Também pode-se notar uma diminuição significativa na quantidade de código gerado, visto que os comportamentos puderam ser reutilizados, com prioridades diferentes em diferentes tipos de agentes, sem necessidade de se duplicar código ou fazer implementações específicas. Foram criados três tipos diferentes de NPCs, com pilhas de comportamentos distintas, que foram organizados em duas equipes. Entre os comportamentos básicos desenvolvidos inicialmente, encontram-se:

**Esperar:** Definido como comportamento padrão de um agente, consiste em manter-se imóvel à espera de que algum comportamento da pilha com prioridade maior torne-se válido;

**Perseguir:** Na presença de um inimigo, o agente o persegue;

**Atirar:** Na presença de um inimigo que esteja próximo o suficiente, o agente dispara contra ele;

**Defender:** Na presença de um aliado que esteja ameaçado por um projétil, o agente tenta colocar-se entre o projétil e o aliado;

**Esquivar:** Na presença de um projétil inimigo, tenta se distanciar de sua trajetória;

**Curar:** Na presença de um aliado ferido, o agente dispara projéteis curativos em sua direção;

**Fugir:** Na presença de um inimigo, tenta fugir dele até que não esteja mais em seu campo de detecção.

Inicialmente, tais comportamentos haviam sido desenvolvidos através de métodos na classe agente, com lógicas extensas para a tomada de decisão sobre qual sequência de ações deveria ser seguida. Tal abordagem tornava difícil o diagnóstico e a correção dos erros de execução dos agentes, além de tornar o comportamento dos agentes bastante linear.

Pilhas de comportamentos foram montadas para cada tipo de agente, de maneira que participassem de uma batalha. Foram adicionados sensores que possibilitaram a observação de atributos dos próprios NPCs, como sua velocidade, seus pontos de vida e dano de projéteis, entre outros.

Para a criação de tais comportamentos, bem como das pilhas de comportamento, o BOD propõe que se monte primeiro uma descrição de alto nível do que se deseja que o agente faça, lembrando sempre de balancear os três elementos tidos como básicos pela metodologia: “Como”, ou a maneira com que o agente deve executar determinado comportamento; “Quando”, ou a especificação das condições sob as quais o agente deve executar as ações de determinado comportamento; e “O que fazer”, ou quais ações deverão ser tomadas para executar o comportamento.

Os três tipos de NPCs criados foram:

**Agressor:** Busca sempre atacar um inimigo, podendo selecionar um alvo inicialmente e atacá-lo utilizando comportamentos *Perseguir* e *Atacar*, além de comportamentos para garantir sua sobrevivência, como *Fugir*, caso a vida estivesse abaixo de um limiar pré-definido, ou tentar *Esquivar* de um projétil inimigo;

**Defensor:** Ataca apenas quando não há um aliado para ser protegido. Caso contrário, executa o comportamento *Defender*. Movimenta-se sempre de maneira a manter-se bloqueando a linha de tiro entre seu protegido e um inimigo (dentro da implementação realizada, NPCs do mesmo time não são capazes de atingir projéteis uns nos outros);

**Curandeiro:** Procura esconder-se (deixando um aliado sempre em sua linha de visão), a fim de evitar danos, e executa o comportamento *Curar* sempre que necessário.

Muitas revisões dos comportamentos foram realizadas e prioridades diferentes nas pilhas foram testadas, fazendo com que se pudesse então criar um jogo simples (ainda que somente com a intenção de testar o modelo) através da inserção um personagem controlado por um jogador humano em uma das equipes e participando dessa batalha.

Mesmo com resultados interessantes, onde agentes se enfrentavam em uma batalha acirrada e executando papéis distintos, apenas ainda podia-se identificar bastante potencial no modelo, uma vez que os agentes ainda eram puramente reativos e mesmo cooperando de certa maneira, não trocavam mensagens entre si, eliminando a possibilidade de classificar o sistema como multi-agente.

## 5 Agentes Planejadores

Até esse ponto o modelo descrito não seguia à risca a metodologia proposta. A pilha de comportamentos descrita deveria sofrer então algumas modificações, passando a possuir estruturas mais elaboradas do que seus comportamentos (a tripla antes descrita) que serão chamadas de BRP (Base Reactive Plan).

Dentro da BRP, novos controles são inseridos, descritos pelo chamado POSH (Parallel Ordered Slip-Stack Hierarchy).

O POSH consiste no conjunto de estruturas que permitem esse chamado planejamento reativo. Tal capacidade de planejamento difere do agente puramente reativo em si no ponto em que as reações são guiadas pelo plano, trocando de reações de acordo com o estado analisado do ambiente.

Os comportamentos inseridos na BRP podem ser chamadas para outras BRPs, com ações e avaliações diferentes, e um controle de quantas vezes cada comportamento foi executado é implementado. Outro fator importante é o fato dos comportamentos retornarem um verdadeiro ou falso, indicando se pela avaliação do agente, o comportamento foi realizado com sucesso ou não, fornecendo um feedback para o agente de suas ações, mecanismo essencial para o planejamento.

Outras estruturas para maior controle também são apresentadas. As que foram utilizadas para testes incluem as “Competences”, que consistem num conjunto de comportamentos que só são verificados caso uma condição anterior tenha sido validada. No exemplo do modelo implementado, a proximidade de um alvo do agressor aciona o conjunto de ações de perseguir, obter distância para atacar e atacar. Caso não seja possível seguir o alvo, ou o ataque seja realizado com sucesso, a “Competence” se encerra. Uma outra estrutura utilizada muito próxima da “Competence” são as “Drive-Collections”, que diferem apenas no ponto em que seus comportamentos são sempre testados, ela não encerra como a “Competence”.

Aplicando as modificações ao modelo anteriormente definido aumenta-se bastante a complexidade da implementação, principalmente pela necessidade do mecanismo de feedback, e porque definições do quão bem uma ação foi executada podem ser muito subjetivas. Mas o trabalho nesse ponto produziu resultados muito superiores aos vistos anteriormente, removendo ainda mais a linearidade das ações dos agentes. Seus comportamentos ainda são previsíveis, mas a gama de comportamentos diferentes executados e a variedade de sua ordem de execução aumentou.

Sem a necessidade de criação de controles específicos, os agentes passaram a ser capazes de selecionar ações de maneira muito mais eficiente. Por exemplo, um agressor com a ação de seguir e atacar tornou-se capaz de perceber falhas em seus ataques (como um inimigo muito rápido que não pode ser atingido) e mudar sua estratégia (passando para um comportamento de menor prioridade na pilha, mas que pode se adequar a situação). Outra adição importante foi o fato de que, nesta versão, comportamentos cujas condições nunca forem atendidas possuem a possibilidade de serem executados a despeito disso, mecanismo poderoso em diversos casos.

Com essa melhora na seleção de ação, obteve-se um planejamento chamado pela autora J. J. Bryson de planejamento reativo, uma vez que o agente ainda possui as características de ter uma pilha de comportamentos que são desencadeados por condições. Mas agora os comportamentos seguem um plano, instituído pela descrição de alto nível exposta anteriormente.

As possibilidades com esses comportamentos continuam a ser exploradas dentro da implementação.

## 6 Multi-Agentes

O BOD não propõe diretamente um tratamento para um sistema multi-agentes, mas, baseado em suas estruturas, tal evolução pode ser mais simples de ser alcançada.

Cada agente se comporta baseando-se em sua memória interna de curto prazo, atualizada pelos seus sensores. Dessa forma, cada agente torna-se um sistema individual, que analisa o meio em que está inserido e interage com ele. Seguindo o BOD, para estabelecer comunicação entre os agentes, sensores e atuadores para viabilizar as mensagens devem ser criados.

A partir desse ponto, fica necessário apenas a adição de comportamentos baseados nas mensagens para criar diferentes ações, como envio, recepção e interpretação dessas mensagens.

Dentro da simulação, isso viabilizou comportamentos como um personagem pedindo ajuda em determinadas ocasiões e atender esse tipo de requisição de outros personagens.

Um controle que foi adicionado nessa simulação ilustra bem o potencial do BOD pois, divididos em comportamentos, podemos elaborar um algoritmo de eleição para possibilitar tomadas de decisões conjuntas de uma equipe de agentes segundo um líder. Tal algoritmo foge ao escopo deste artigo, mas foi baseado amplamente em um algoritmo de eleição utilizado em sistemas distribuídos para eleições em sistemas operando sobre redes sem fio (veja [Tanenbaum and Steen 2008]), viabilizado graças aos mecanismos do BOD e à abstração proporcionada por ele e pela POO, de que cada agente poderia ser tratado como um sistema distinto com suas próprias versões de sensores e atuadores. Se pensarmos sempre dessa forma, fica fácil extrapolar a ideia de um ambiente computacional síncrono e único (i.e. um jogo com diversos agentes simulados num computador), para um ambiente assíncrono em que cada agente seria executado, por exemplo, em um robô distinto, e um grupo deles agiria em conjunto para derrotar um jogador ou mesmo superar labirintos físicos, trazendo novas possibilidades de interação utilizando realidade aumentada e/ou outras aplicações, cuja implementação seria facilitada através da metodologia estudada.

## 7 Conclusões

Procuramos ao longo deste artigo apresentar o potencial do BOD quando voltado para o desenvolvimento de uma Inteligência Artificial para jogos. Procuramos mostrar os pontos em nossas simulações, uma vez que consideramos os resultados muito diferentes do normal encontrado em jogos.

Nossas simulações foram testadas apenas em um cenário. Assim, para dar continuidade ao trabalho pretendemos enriquecer esses testes com funcionalidades diferentes, além de modelar melhor os comportamentos para explorar os usos do POSH.

Pretendemos também utilizar mais algoritmos de IA tradicionais, uma vez que esses podem ser inseridos como objetos, de forma modularizada. Acreditamos assim que as possibilidades são infinitas, até mesmo trabalhos de outros ramos da computação podem ser testados, e a aplicação do BOD não se limita apenas a jogos (a própria autora dá finalidades bem diferentes para o método, utilizando por exemplo para simular comportamentos de mamíferos).

## Referências

- BRYSON, J. J. 2001. *Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents*. PhD thesis, MIT, Department of EECS, Cambridge, MA. AI Technical Report 2001-003.
- BRYSON, J. J. 2003. The Behavior-Oriented Design of modular agent intelligence. In *Agent Technologies, Infrastructures, Tools, and Applications for e-Services*, R. Kowalszyk, J. P. Müller, H. Tianfield, and R. Unland, Eds. Springer, Berlin, 61–76.
- GREY, J., AND BRYSON, J. J. 2009. Procedural quests: A focus for agent interaction in role-playing-games. In *Proceedings of the AISB 2011 Symposium: AI & Games*, SSAISB, York, D. Romano and D. Moffat, Eds., 3–10.
- RUSSEL, S., AND NORVIG, P. 2003. *Artificial Intelligence: A Modern Approach*, 2 ed. Prentice Hall.
- TANENBAUM, AND STEEN, V. 2008. *Sistemas Distribuídos: Princípios e Paradigmas*, 2 ed. Prentice-Hall.