Hidra: uma Inteligência Artificial Solucionadora de Labirintos

Guilherme N. M. Daudt, Fábio Y. Okuyama[1]

[1]Sistemas para Internet - Instituto Federal do Rio Grande do Sul (IFRS)
Caixa Postal xxx - 90.035-007 - Porto Alegre - RS - Brasil

{gdaudt@gmail.com, fabio.okuyama@poa.ifrs.com.br}

Resumo. *Este artigo apresenta a descrição do desenvolvimento e funcionamento do agente de inteligência artificial Hidra e a comparação entre duas estratégias de resolução do problema proposto, motivado pela inteligência artificial presente nos jogos. Visando tornar suas ações semelhantes às de um ser humano, foram usadas técnicas de aprendizado por reforço em seu desenvolvimento. O ambiente de atuação escolhido foi o mundo Wumpus. Dentro deste mundo, foram desenvolvidas duas abordagens diferentes para a tentativa de resolução do problema. A primeira abordagem faz com que o agente tome ações mais concretas. Na segunda abordagem ele executa ações mais abstratas, analisando o ambiente e a sua situação no momento da ação. Na segunda abordagem foram utilizadas técnicas de Redes Bayesianas para auxiliar no mapeamento e inferência de obstáculos. No fim do artigo, são apresentados os resultados de ambas.*

1. Introdução

A inteligência artificial em jogos está presente praticamente desde sua criação. Os primeiros jogos, como *Tennis for Two* e *Spacewar!* foram desenvolvidos para dois jogadores, em função das limitações tecnológicas da época [Kent, Steven L.]. O jogo *Pac-Man* foi um dos jogos populares a apresentar mais cedo um tipo de inteligência artificial mais complexa. No entanto, a base da inteligência presente nos fantasmas do *Pac-Man* não era completamente diferente das inteligências presentes em jogos de 15 depois de sua criação [Millington, I. e Funge, J. 2009]. Hoje em dia, com a evolução dos jogos em um nível cada vez mais realístico, falhas na inteligência das personagens, sendo amigos ou inimigos, ficam cada vez mais evidentes e mais criticadas. Tendo em vista os fatores citados, foi desenvolvido um sistema com o uso de algumas técnicas de IA com o objetivo de interagir com o ambiente e tirar proveito disso, denominamos este sistema Hidra. A metodologia escolhida foi a de aprendizado por reforço que, segundo pesquisas feitas até o momento, não é uma técnica comumente utilizada em jogos comerciais. Trabalhos envolvendo jogos e aprendizado por reforço podem ser encontrados em [Amato, C. e Shani, G], [McPartland, M. e Gallagher, M] . O cenário de teste para Hidra é uma versão adaptada do problema do mundo Wumpus [Thielser, M]. O objetivo é que o desenvolvimento deste sistema permita posteriormente a generalização da abordagem com a criação de uma biblioteca ou framework

para jogos e contribua para difusão desta abordagem nos modelos de inteligência dos jogos modernos.

Nas próximas seções do artigo serão apresentados, respectivamente, o mundo Wumpus e uma breve explicação sobre sua apresentação e funcionamento, alguns conceitos de aprendizado por reforço que serviram como base para a confecção do projeto e uma explicação suscinta sobre Q-learning, a seção do aprendizado por reforço utilizada e sobre redes bayesianas. Também será apresentado o conceito da Hidra, assim como sua implementação e a conclusão alcançada através de todo o processo.

2. O mundo Wumpus

O mundo Wumpus foi apresentado pela primeira vez por Michael Genesereth, e consiste de um universo pequeno, com um agente e obstáculos previamente estabelecidos. O labirinto original foi desenvolvido a partir de uma matriz (4,4), e possui dois obstáculos. O obstáculo principal, que dá nome ao problema, é o Wumpus, um monstro comedor de gente, e se o agente chegar na mesma posição que o Wumpus, é comido imediatamente. O segundo obstáculo são poços, nos quais ele morre se caminhar para o mesma posição destes. A única maneira dele de evitar cair nestas armadilhas é recorrer a sua base de conhecimento, previamente construída. Um exemplo de mapa está exemplificado na figura 1. O que ele sabe é que nas posições adjacentes a um poço existe uma brisa, e nos quadrados adjacentes ao Wumpus, existe um mau-cheiro.

Para progredir no labirinto, o agente pode executar uma série de ações. Ele pode virar para esquerda ou para direita, ir para frente, pegar, soltar, atirar e sair da caverna, se estiver na posição inicial (1,1). Ele possui uma flecha, que é atirada na direção em que o agente estiver virado. Se a flecha acertar o Wumpus, este é morto imediatamente, e ele emite um grito que pode ser ouvido por toda a caverna (significando que o mesmo perceberá quando o Wumpus foi morto ou não). Senão, ela segue até o final do labirinto e é absorvida pela parede. O agente percebe um brilho quando está na mesma posição do ouro, e também percebe quando tenta em um espaço inválido e bate em uma parede.

O agente inicia o labirinto na posição (1,1) e virado para o leste, e seu objetivo é percorrer o labirinto, pegar o ouro e sair da caverna com ele. A avaliação do algoritmo é feita a partir da performance deste no labirinto. Ele recebe +1000 pontos por pegar o ouro, -1000 por cair em um poço ou ser comido pelo wumpus, -1 para cada ação tomada e -10 por usar a flecha.[Russel, S. and Norvig, P.]. A proposta de resolução do problema original foi fornecer previamente uma base de conhecimento ao agente para que este tivesse a capacidade de identificar os obstáculos do mapa e chegar ao seu objetivo sem cometer nenhum erro, e voltar com segurança ao início. A abordagem proposta neste artigo, no entanto, consiste em fazer com

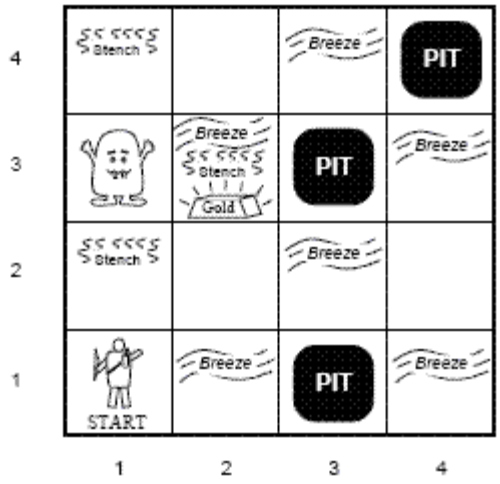

Figura 1 - O mundo Wumpus

que o agente construa a sua própria base de conhecimento, utilizando as técnicas de aprendizado por reforço. Através da exploração e experimentação de novos caminhos, a idéia é que as experiências tanto de sucesso quanto de falha, sejam usadas para aquisição de conhecimentos a respeito dos problemas que está enfrentando, e que, após algumas tentativas e erros, alcance um resultado semelhante a um que possua uma base previamente estabelecida. A definição e exemplificação de aprendizado por reforço será apresentada na próxima seção.

3. Aprendizado por Reforço

Pensando de uma maneira essencial, as pessoas aprendem através de suas experiências [Kolb, D, 1999]. Quando são crianças, interagem com o seu ambiente sem nenhum tipo de orientação, e ainda assim conseguem tirar proveito dos acontecimentos. Através dessa afirmação, é possível afirmar também que estas interações são a sua maior fonte de saber, seja sobre o ambiente ou sobre suas próprias limitações e capacidades. “Aprender através de uma interação é a idéia fundacional por baixo de quase todas as teorias de aprendizado e inteligência”[Sutton e Barto 2005]. O importante para que se atinja o resultado esperado na resolução do problema apresentado no artigo é que consiga se transformar este conceito, que antigamente se aplicava apenas aos humanos, em uma abordagem computacional, conseguindo fazer com que um agente aprenda da mesma forma que uma pessoa.

Segundo Richard S. Sutton e Andrew G. Barto, aprendizado por reforço é saber o que fazer - como mapear situações para ações - para maximizar um sinal numérico de recompensa. Os autores também afirmam que o aprendizado por reforço não é definido pela caracterização de uma metodologia de aprendizado, apesar do que pode parecer, e sim pela caracterização de um problema de aprendizado. O que desafia a inteligência que trabalha com aprendizado por reforço, e apenas neste tipo de inteligência, é a recompensa atingida a partir da troca de explorações com o ambiente. Tradicionalmente, o agente trabalha com a obtenção de recompensas para avaliar o

seu desempenho. A fim de alcançar as melhores recompensas, ele deve ser capaz de tirar proveito do seu conhecimento adquirido. Ao mesmo tempo, também precisa explorar o ambiente em partes ainda desconhecidas, para fazer novas descobertas e expandir o seu conhecimento.

Para isso, o mesmo se apóia nas quatro bases que compõem a metodologia[Sutton e Barto 2005]. A primeira é a Política, que consiste das ações que devem ser tomadas quando enfrentadas determinadas situações. Seria a resposta dada à ocorrência do contato do agente com algum elemento do ambiente, seja ele conhecido ou não. Para uma política ser formada, é preciso utilizar um sistema de recompensas, segundo ponto da base. As recompensas avaliam as ações imediatas tomadas por ele através de um valor numérico, determinando se estas são boas ou ruins. O terceiro ponto é o sistema de valores, que, ao contrário do sistema de recompensas, avalia os conjuntos de ações, pois uma ação isolada pode possuir uma recompensa pequena, mas o conjunto que ela desencadeia leva a uma recompensa total maior. Para este utilizar de forma otimizada estes três pontos, é necessário o quarto, o modelo. O modelo é a parte responsável pelo planejamento das ações, fazendo simulações do comportamento do ambiente, diferenciando dos agentes de aprendizado por reforço antigos, que funcionavam quase que exclusivamente com tentativa e erro [Kaelbling, L., Littman, M. e Moore, A].

4. Q-Learning

O Q-learning é uma parte especial do aprendizado por reforço, pois trabalha de uma maneira singular o uso de ações e recompensas. Esta técnica vincula os estados possíveis do operador com as ações que este pode tomar. Os estados são relativos a qualquer tipo de percepção que ele tenha do ambiente, como sensações e posição do mesmo. É utilizada a premissa de que exista um número limitado de ações a serem tomadas em relação aos finitos estados presentes no ambiente - por exemplo, no problema do Wumpus o agente é limitado às ações citadas anteriormente, como ir para frente, pegar, atirar e outras - é possível concluir que cada ação tomada dentro de cada determinado estado gerará uma recompensa diferente, e através destes valores de recompensa que o agente consegue construir sua estratégia.

A ideia de Q-learning foi apresentada pela primeira vez por [Watkins, 1989], que projetou que um operador que "aprenderia estimando inicialmente um valor *Q*, e dados através da experiência, que consiste de observações da forma (consultar equação na referência) onde *x* representa o estado, *a* representa a ação tomada no estado, *r* a recompensa imediata recebida e *y* o estado subsequente alcançado". Desta maneira, a proposta é que o valor de *Q* possa ser estimado através das observações feitas levando em conta os quatro elementos *x*, *a*, *r* e *y*. O valor destas observações é calculado através de algumas equações, que sofreram variações a medida que a técnica foi estudada. Um elemento em comum em quase todas suas variações é o α, que seria o quociente de aprendizado do agente. Este quociente possui valores entre 0 e 1, e pode ser modificado de acordo com a progressão do agente. O valor desse quociente determina o quanto que o agente irá aprender. Quanto mais próximo o valor utilizado for de 0, significa que haverá menos consideração para recompensas imediatas, enquanto um valor próximo de 1 fará com que ele

leve em consideração apenas os resultados mais atuais. O elemento **γ,** também comum, é um quociente que tem uma função oposta ao α, sendo chamado de taxa de desconto. Este quociente determina o quanto das ações passadas serão consideradas. Sendo próximo de 1, considera muito as ações passadas, sendo próximo de 0, desconta maior parte de seus valores. Uma das opções de calcular a atualização de valores em uma Q-table está apresentada na equação 1.

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma max_{a'} Q[s',a'] - Q[s,a])$$

Equação 1. Atualização do valor das recompensas na tabela

5. Redes Bayesianas

As redes bayesianas são estruturas gráficas utilizadas para representar graficamente as probabilidades de um domínio que não é conhecido. São geralmente dispostas através de grafos. "De uma maneira informal, as redes bayesianas são grafos acíclicos diretos (DAG), onde os nodos são variáveis aleatórias, e os arcos especificam a independência das inferências que ocorrem entre as variáveis aleatórias"[Larrañaga, P., Posa, M., Yurramendi, Y., Murga, R., Kuijpers, C.]. Um DAG é uma estrutura que possui um conjunto de nós e de pontas. Os nodos correspondem às variáveis, e são representados através de um círculo e nomeados de acordo. As pontas representam a dependência direta entre as variáveis, e são representadas através de setas entre os nodos[Thulasiraman, K.; Swamy, M. N. S.]. Por exemplo, com uma ponta vinda do nó A para o nó B significa que existe uma dependência estatística entre as duas variáveis. Isso significa que o valor de B é influenciado pelo valor de A, podendo ser resumido a ponto de dizer que A é pai de B e e B é filho de A.

Dentro do mundo Wumpus, as redes bayesianas são utilizadas para guiar o caçador através do mapa, tentando evitar que ele encontre perigos, tirando vantagem de suas inferências a partir das sensações recebidas pelo mesmo. Trabalhos que utilizaram os conceitos de redes bayesianas para a solução do problema Wumpus podem ser vistos em [Parker, L.], [Provan, G].

6. Hidra

Este trabalho propõe um agente de Inteligência Artificial capaz de aprender com as suas próprias experiências. A proposta é desenvolvê-lo utilizando as técnicas de aprendizado por reforço (Q-learning) e o auxílio de redes bayesianas para a resolução do problema Wumpus. Para isto, foram determinadas duas abordagens: Uma delas faz uso de uma visão mais concreta do problema, na qual ele utiliza ações absolutas semelhantes ao que seriam se fosse aplicada outra técnica de inteligência artificial, sem o auxílio das redes bayesianas. A segunda tenta ser um pouco mais abstrata, com ações originais propostas a fim de explorar as probabilidades vindas das inferências das redes bayesianas. A personagem principal do problema será um robô, enviando sondas, tendo

como objetivo explorar o labirinto apresentado a ele, adquirindo as melhores recompensas possíveis.

A figura do robô foi escolhida pois, em função de sua natureza exploratória por possuir um raciocínio baseado em recompensas, é esperado que ele falhe algumas vezes até que o conhecimento construído seja suficiente para que suas sondas consigam resolver o labirinto sem a ocorrência de falhas. Sendo cada falha cometida fatal, assim como no problema do Wumpus, imagina-se que as sondas seriam destruídas.

Desta maneira, tentando manter uma perspectiva real, ainda que simulada virtualmente, quando uma sonda fosse destruída uma nova seria reconstruída, e o robô carregando toda a base de conhecimento adquirida com a sonda anterior, fazendo com que ao final da simulação fosse construído um modelo ideal para a resolução do problema. O nome Hidra é uma analogia à criatura da mitologia grega, que era capaz de regenerar suas cabeças quando estas eram cortadas. O corpo representaria o robô, capaz de construir as sondas, e as cabeças seriam representadas pelas sondas, pois quando uma delas for destruída, outra será formada no seu lugar.

A aplicação das redes bayesianas na Hidra é fazer uso das probabilidades para auxiliar sua movimentação através do mapa e mapear de maneira segura os obstáculos a serem evitados durante o seu trajeto.

O objetivo final da Hidra é conseguir fazer com que as ações melhores recompensadas durante a construção de seu conhecimento sejam as ações que alcancem uma pontuação semelhante a agentes que utilizam técnicas diferentes.

7. Implementação

O ambiente no qual a Hidra foi posicionada possui as mesmas características do mundo Wumpus. Logo, suas sondas se movimentarão em um mapa representado por uma matriz(4,4), com dezesseis posições no total. Cada uma dessas posições será interpretada como um estado para elas, portanto também serão dezesseis estados. Em relação aos obstáculos, serão três poços e um Wumpus, em posições diferentes entre si. As sondas partem tradicionalmente da posição (1,1), que se fica no local mais abaixo e à esquerda do mapa, e são retornadas para esta posição a cada vez que executa uma ação que a leva a um estado definitivo, neste caso, a morte ou encontrar o ouro. Para cada ação efetuada, será dada uma recompensa referente ao resultado desta ação. As recompensas referentes aos resultados está descrita no quadro 1. O *Win* ocorre quando uma das sondas consegue alcançar a posição na qual o ouro está localizado. O *Success* acontece quando o agente chega em uma posição válida sem nenhum obstáculo. O *Unavailable* ocorre quando ele tenta andar para uma posição inválida ou erra o seu tiro. O *Killed* acontece quando se chega em uma posição que contém tanto o poço quanto o wumpus e o *Wumpus Killed* quando se acerta o tiro no Wumpus. Cada uma das células da matriz armazena uma Q-table, tabela onde a Hidra armazena as suas recompensas. Esta tabela está ordenada em função das ações, e possui dois campos, o primeiro armazena o valor acumulado da recompensa em função da respectiva ação e

o outro armazena o número de vezes que a ação foi executada. Os valores nesta tabela são atualizados de acordo com a equação 1.

Quadro 1. Valores de recompensas para os resultados

Win	Success	Unavailable	Killed	Wumpus Killed
+500	+50	-50	-500	+250

A primeira abordagem descrita será a concreta. Nela, as ações propostas são ações semelhantes à movimentação no mundo Wumpus tradicional. São elas *Move up, Move down, Move left, Move right, Shoot up, Shoot down, Shoot left, Shoot right*, formando nesta ordem uma Q-table em cada uma das células com oito colunas(número de ações) e duas linhas. O α escolhido nesta situação possui um valor dinâmico, tendo o valor de 1 sobre o número de vezes que a ação foi efetuada naquele estado. A Q-table de uma célula de exemplo está exemplificada no quadro 2. A escolha das ações é determinada de maneira aleatória na sua fase de exploração, determinada arbitrariamente para acabar depois da vigésima iteração (cada iteração sendo contada quando ocorria a morte ou a vitória do wumpus). Após essa fase, começava a fase em que as ações eram determinadas de acordo com os seus valores nos seus respectivos estados. A cada estado enfrentado, era verificado na tabela qual era a ação que obteve o melhor resultado e a mesma era realizada.

Quadro 2. Exemplo de Q-table referente à primeira abordagem

	Move up	Move down	Move left	Move right	Shoot up	Shoot down	Shoot left	Shoot right
Q-value	+50	-50	0	0	0	0	0	0
Times repeated	1	1	0	0	0	0	0	0

Após implementada desta maneira, a Hidra foi submetida a diversas iterações com o mesmo mapa para verificar se as ações com os valores maximizados correspondiam às melhores ações de resolução do mapa ao qual ela foi apresentada. Um exemplo de mapa testado está presente na figura 2. Após 40 iterações foi verificado que no mapa apresentado a Hidra maximizou as ações *Move right, Move up, Move left e Move up*, que é o caminho mais curto e mais seguro para a vitória. A segunda sequência de ações melhor pontuada era de *Move right, Move up, Shoot up, Move up e Move right*, caracterizando uma vitória, mas em um caminho mais longo.

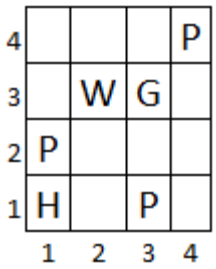

Figura 2. Mapa exemplificado. H representa a sonda, P os poços, W o wumpus e G o ouro.

A segunda abordagem é a abstrata. Nela, as ações escolhidas possuem relação com as suas percepções e as probabilidades de perigo nas suas células adjacentes. Aqui foram utilizados os conceitos de redes bayesianas para fazer o mapeamento das células que eram seguras e as que eram perigosas. O comum quando se está utilizando redes bayesianas é que se leve em conta todos os perigos possíveis e se faça uma inferência em todas as posições do mapa. A medida que o agente se desloca, as probabilidades são atualizadas. No entanto, o objetivo nesta abordagem era dar muito pouco conhecimento à Hidra, então o mapeamento dos perigos foi feito apenas em suas células adjacentes. A medida que suas sondas se deslocavam, verificavam a percepção da posição onde se encontravam e se não houvesse nem brisa nem fedor, marcava todas as posições adjacentes como seguras, assim como a posição onde estava. Se houvesse uma brisa ou um fedor, ela verificava quais das células não haviam sido exploradas, e infere o valor da probabilidade de perigo relativo a sensação igual a 1 sobre a quantidade de células não visitadas, a cada uma das células inexploradas.

As ações propostas são as seguintes: *Wumpus*, onde a sonda vai em direção à célula que possui a maior probabilidade de possuir o Wumpus; *Pit*, onde ela vai em direção à célula que possui a maior probabilidade de possuir um poço; *Risky*, é uma ação na qual ela arrisca uma célula que possui um perigo que é maior do que 0, mas menor do que 1. Se nenhuma dessas condições for atinginda, ela se dirige até a célula menos visitada; *Safe*, ela se dirige a uma célula com nenhuma probabilidade de perigo, preferencialmente já visitada antes; *Shoot*, ela atira uma flecha em direção à posição com a maior probabilidade de conter o Wumpus. As fases de escolha de ação são semelhantes à primeira abordagem.

Após implementada desta maneira, a Hidra foi submetida a testes semelhantes aos feitos na abordagem anterior. Depois de passar por diversas iterações, em mapas diversos, foi verificado que as sondas conseguiam com sucesso mapear os perigos presentes no mapa, no entanto, não conseguiam maximizar as ações que as levariam à vitória. Isto ocorre em função do conceito por trás de cada ação, que não garante que uma ação efetuada em determinada posição resulte na sonda indo para a mesma posição todas as vezes.

8. Conclusão

Após a implementação de ambos os tipos e dos testes efetuados, foi possível concluir que, da maneira que a Hidra foi modelada, ações mais concretas facilitam a construção de uma política, pois o agente possui uma certeza de que a ação tomada o levará sempre ao mesmo resultado. No caso da segunda abordagem, seria necessário montar um plano de política mais complexo do que o proposto, para que as sondas fossem capazes de avaliar o conjunto das ações ao invés de considerarem ela individualmente, sendo assim, possível sacrificar uma das ações pelo benefício do conjunto.

9. Referências

Kolb, D., Boyatzis, R., Mainemelis, C. (1999). "*Experiential Learning Theory: Previous Research and New Directions*". Disponível em: <http://www.d.umn.edu/~kgilbert/educ5165-731/Readings/experiential-learning-theory.pdf>. Acesso em: 05 jul. 2013;

Kaelbling, L., Littman, M. e Moore, A. (1996). *Reinforcement Learning: A Survey*. Disponível em:<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/kaelbling96a-html/rlsurvey.html>. Acesso em: 05 jul. 2013.

Kent, Steven L. (2000). *The First Quarter: A 25-year history of video games*. BWD Millington, I. e Funge, J. (2009). *Artificial Intelligence for Games*. Elsevier Inc.

Russel, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education Inc.

Sutton, R. e Barto, A. (2005). *Reinforcement Learning: An Introduction*. Disponível em: <http://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html>. Acesso em: 25 jan. 2013.

Thielser, M. *Designing a FLUX Agent for the Dynamic Wumpus World.* Disponível em: <http://www.computational-logic.org/content/projects/wisslogpubs/NMR06Thielscher.pdf>. Acesso em: 30 jan. 2013.

Watkins, C.J.C.H., (1989), Learning from Delayed Rewards. Ph.D. thesis, Cambridge University

Woergoetter, F. e Porr, B. (2007). "*Reinforcement Learning".* Disponível em: <http://www.scholarpedia.org/article/Reinforcement_learning>. Acesso em: 30 jan. 2013

Thulasiraman, K. e Swamy, M. N. S. (1992), "5.7 Acyclic Directed Graphs", *Graphs: Theory and Algorithms*, John Wiley and Son, p. 118

Provan, G. (2009) *Course:* "*Introduction to Artificial Intelligence*". Disponível em:

<http://www.cs.ucc.ie/~gprovan/CS3315-FY08/Supplementary%20Notes.pdf>. Acesso em: 05 jul. 2013.

Parker, L. (2004) *Course: "Artificial Intelligence*". Disponível em:

<http://web.eecs.utk.edu/~parker/Courses/CS594-fall04/Lectures/chapter13-14parts.pdf>. Acesso em: 05 jul. 2013.

Amato, C. e Shani, G. (2010) "*High-level Reinforcement Learning in Strategy Games*". Disponível em:

<http://people.csail.mit.edu/camato/publications/LearningInCiv-final.pdf>. Acesso em: 05 jul. 2013.

McPartland, M. e Gallagher, M. (2010) "*Reinforcement Learning in First Person Shooter Games"*. Disponível em:

<http://elec.uq.edu.au/~marcusg/papers/mcpartland_gallagher_tr_ciaig.pdf>. Acesso em: 05 jul. 2013.