

Aplicação de avatares autónomos para desempenhar o papel de membros na execução de trabalhos em equipa

Andreas Vilela

ECT/UTAD – Universidade de Trás-os-Montes e Alto Douro
Departamento de Engenharias
Vila Real, Portugal
andreasvilela@gmail.com

MAJ Aires Marques, SAJ Henrique Costa, TEN Jorge Rafael

Esquadra de Manutenção de Motores da Base Aérea N.º 5
Serra de Porto de Urso, Monte Real, Portugal
airestm@gmail.com, kosuta2@hotmail.com,
tenjorgerafael@gmail.com

Rui Prada

IST – Instituto Superior Técnico
Lisboa, Portugal
rui.prada@ist.utl.pt

Leonel Morgado

GECAD/UTAD – Universidade de Trás-os-Montes e Alto Douro
Vila Real, Portugal
leonelm@utad.pt

Resumo – A possibilidade de treino de uma tarefa efetuada em equipa exige a disponibilidade de horário e de homogeneidade de preparação de todos os membros do processo de treino, o que limita as possibilidades de treino individual de uma tarefa, mesmo que em ambiente virtual. A aplicação de avatares autónomos a estes cenários, em substituição de alguns dos elementos humanos da equipa, pode permitir ultrapassar estas limitações, para que o treino possa decorrer mesmo que só parte dos membros humanos esteja disponível ou preparada. Apresentamos uma proposta e protótipo neste sentido, aplicada ao caso concreto da formação de mecânicos da Força Aérea Portuguesa em operações de manutenção de motores de aeronaves F-16, partilhando um mesmo espaço virtual 3D em OpenSimulator.

Palavras-chave: Agentes inteligentes, avatares autónomos, formação, mundos virtuais 3D, tarefas em equipa, OpenSimulator, F-16.

I. INTRODUÇÃO

O processo de treino de tarefas em equipa implica a conjugação das disponibilidades de horário dos integrantes da equipa. Também o nível de preparação prévia de cada membro afeta a forma como pode decorrer um exercício em equipa. Estes aspetos conferem alguma complexidade logística ao treino de uma tarefa em equipa, quer em contexto presencial, quer em contexto virtual.

Contudo, considerando o caso do treino de tarefas de equipa em contexto virtual, onde cada membro da equipa se apresenta sob a forma de um avatar, idealizámos que alguns desses avatares pudessem agir como agentes inteligentes. Na nossa perspetiva, numa situação em que só houvesse disponibilidade para participação no processo de treino por parte de algumas pessoas, o treino em equipa ainda assim pudesse decorrer para essas, com os papéis dos membros ausentes assumidos por avatares controlados por um sistema de decisão. A nossa opção pela designação “avatar autónomo” em vez de “agente autónomo” pretende dar ênfase à flexibilidade antevista para o número de seres humanos na equipa: do ponto de vista dos utilizadores, há quatro avatares em cooperação, podendo quaisquer deles ser controlados por um sistema inteligente ou por seres humanos.

Este artigo apresenta um protótipo que explorou a possibilidade de concretização desta proposta no caso de um problema concreto: o treino de manutenção mecânica em espaço virtual 3D na Força Aérea Portuguesa (FAP). Nomeadamente no processo de instalação do motor Pratt & Whitney F100 numa aeronave F-16 [1].

O estudo efetuado neste caso concreto analisou as tarefas deste processo, extraindo delas um conjunto de ações e condições que permitissem efetuar a automatização individual de cada interveniente na tarefa. Para realizar o comportamento autónomo recorreu-se ao algoritmo *Partial-Order Planning* [2] transformando as ações identificadas em operadores do algoritmo de planeamento.

II. CASO DE ESTUDO

Na Força Aérea Portuguesa (FAP), a formação básica dos Mecânicos de Material Aéreo (MMA) é efetuada no Centro de Formação Militar e Técnica da Força Aérea, na Ota. Após esta formação básica, os MMA são destacados para outras unidades, direções ou órgãos da FAP, onde recebem então formação específica para os motores em que vão atuar. É o caso da Base Aérea N.º 5 (BA5), em Serra do Porto de Urso, Monte Real, Leiria, a partir da qual operam as aeronaves F-16 da FAP. Na BA5, a formação dos MMA integra fases teóricas e fases práticas, efetuadas sobre componentes mecânicos reais. Desta forma, a componente prática da formação requer o envolvimento de recursos físicos (aviões e motores) fazendo com que estes não possam estar disponíveis durante esse processo. Assim, estes momentos de formação tornam-se particularmente dispendiosos.

Neste contexto, está em curso uma cooperação entre a FAP e a Universidade de Trás-os-Montes e Alto Douro (UTAD) para desenvolvimento de um ambiente de formação virtual 3D que pretende contribuir para aumentar a eficiência destes momentos de formação, permitindo aos formandos a possibilidade de praticar as tarefas de manutenção e os aspetos de coordenação entre membros da equipa de forma simulada, acessível a qualquer momento, sem envolvimento de recursos físicos – com a ambição de

umentar a preparação prévia dos formandos para a posterior formação prática com aviões e motores reais [3].

O presente protótipo focou-se na primeira tarefa focada por esse ambiente em desenvolvimento, o já referido processo de instalação do motor F100 numa aeronave F-16.

Nesta tarefa, o processo de treino segue um conjunto de várias tarefas de execução, que podem ser ou não interdependentes. Como é comum na manutenção em mecânica aeronáutica, estes processos são especificados pelos fabricantes através de documentos designados “*Technical Orders*” (TO), que descrevem as várias fases das tarefas, através de texto e diagramas [1]. No âmbito da colaboração em curso entre a UTAD e a FAP, esta descrição foi analisada e descrita como um processo multiagente. Esta análise foi complementada com registos de vídeo e imagem dos processos reais, e transformada num guião multiagente [4].

Este desdobramento da TO de tarefas em guião multiagente foi o nosso ponto de partida para tentar utilizar avatares autónomos em substituição de seres humanos na execução de tarefas em equipa. Dado termos nesse guião a identificação clara de quais os papéis a desempenhar por cada avatar numa tarefa em grupo (com mais que um mecânico simultaneamente), torna-se possível definir um plano ordenado das ações e condições necessárias ao desenrolar o processo, aspeto que suporta a aplicação de *Partial-Order Planning* no desenvolvimento dos agentes inteligentes. Desta forma, constatámos estar perante uma oportunidade para tentar desenvolver um protótipo que concretizasse a ideia original (equipas mistas de avatares autónomos e avatares controlados por seres humanos).

III. ANÁLISE DE TAREFAS

O processo de análise e representação é muito importante para se desenvolver uma boa representação do sistema e consequentemente um bom planeamento. Dada a extensão do conjunto de tarefas associadas à instalação do motor F100 num F-16, optámos por nos focar em alguns casos concretos dentro destas. Apresenta-se nesta secção a análise desses casos concretos de tarefas.

O processo de treino em análise, em contexto real de formação, exige a presença de três ou quatro mecânicos, organizados em redor do motor (Figura 1).



Figura 1 – Mapa de localização de cada mecânico.

A primeira tarefa do processo consiste em remover quatro parafusos da fuselagem do avião F-16. Trata-se de uma tarefa que pode ser efetuada isoladamente, mas também em grupo, tendo os quatro mecânicos um determinado papel na mesma. A tarefa é constituída por quatro ações, sendo que numa perspetiva de trabalho em grupo cada mecânico terá a responsabilidade de executar uma. Cada uma destas ações tem por objetivo remover um parafuso; a execução de cada uma terá assim associada uma alteração no estado do sistema. Como são ações idênticas, representámo-las por uma ação genérica de remoção de parafuso, a aplicar 4 vezes, uma por cada parafuso. A figura 2 explicita esta ação genérica. Neste caso, esta ação desempenhada por um determinado mecânico “mec” permite que um parafuso “p” passe do estado Apertado para Desapertado em determinado *slot* “s”, ou seja, a variável do tipo parafuso sofrerá alterações ao nível do seu estado passando de Apertado para Desapertado. Esta ação apresenta uma pré-condição e uma pós-condição, que definem as alterações de estado do sistema antes e depois da execução da mesma. Posteriormente, o algoritmo com base nos estados do sistema irá gerar um conjunto de ações para satisfazer as tarefas do processo.

Na Figura 3 apresenta-se o caso concreto em que a aplicação desta ação dedica a cada mecânico a tarefa de remoção de um parafuso diferente.

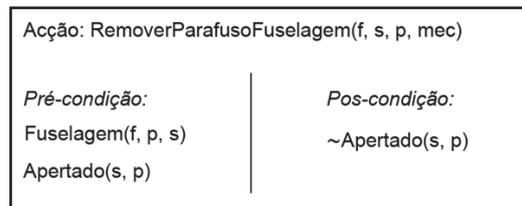


Figura 2 – Ação genérica para remover um parafuso.

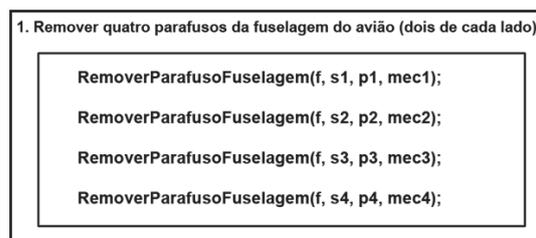


Figura 3 – A ação genérica aplicada quatro vezes, para remover os quatro parafusos da fuselagem do avião.

Desta forma, a ação de remover o parafuso 1 terá como pré-condição “Apertado(s1, p1)” e como pós-condição “~Apertado(s1, p1)”. Assim, depois de esta ação ser executada corretamente, o parafuso “p1” deixará de estar apertado no *slot* “s1” da fuselagem do avião.

Nesta abordagem, em caso de execução incorreta de uma ação por parte de um mecânico, as ações associadas aos outros mecânicos não são afetadas. Desta forma, em caso de intervenção incorreta de um aluno/formando ou mesmo inação, o sistema pode encontrar o ponto concreto em falha. Por sua vez, uma tarefa de grupo só é considerada concluída quando todas as ações e ela associadas forem executadas corretamente. No entanto, grande parte das tarefas não exigem sincronismo na execução das mesmas por parte dos utilizadores ou avatares autónomos. Além disso, só se prossegue para a tarefa seguinte quando se conclui uma tarefa de grupo por completo, pois todas as partes intervenientes no sistema comunicam com um serviço externo para se manter sincronização de estados.

Outra tarefa do processo consiste em instalar três apoios no carro de instalação do motor na aeronave. Tal como a tarefa anterior, esta tarefa pode ser executada em grupo pelos mecânicos intervenientes. Os três apoios a instalar destinam-se a elevar o motor para posterior introdução na aeronave.

Também neste caso é possível definir uma ação genérica de instalação de um apoio (Figura 4) e posteriormente aplicar esta tarefa aos apoios a instalar (Figura 5).

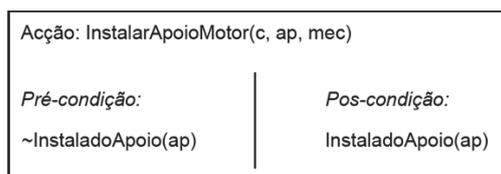


Figura 4 – Ação genérica para instalar um apoio no motor.

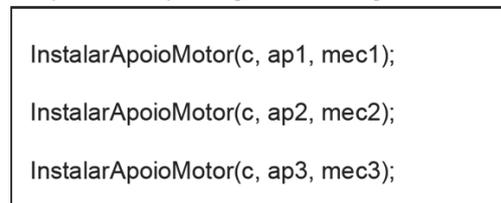


Figura 5 – Instalação de três apoios no motor.

Cada ação da Figura 5, quando executada, permite alterar o estado de um apoio no sistema. A ação é executada por um mecânico (“mecX”), fazendo com que um apoio passe do estado “~InstaladoApoio(ap)” para “InstaladoApoio(ap)”, sendo a variável “apX” um apoio do motor.

Como se referiu anteriormente, o processo de instalação do motor na aeronave F-16 pode ser efetuado por quatro mecânicos. Nesta tarefa específica, três instalam os apoios (um por mecânico) e o quarto mecânico supervisiona a tarefa, verificando posteriormente se a instalação dos componentes foi executada corretamente. Esta tarefa de

supervisionamento, efetuada pelo quarto mecânico, é uma ação seguinte no processo.

Tal como acontece no processo real de formação, uma determinada tarefa não tem que ser necessariamente executada por determinado mecânico, no entanto, a tarefa tem que estar concluída. Neste sistema implementado, o avatar autónomo executa apenas a tarefa que lhe é atribuída, no entanto, se um utilizador real executar uma tarefa atribuída a um avatar autónomo nada no sistema é afectado e o treino procederá normalmente.

Em suma, a análise do processo para execução com avatares autónomos visa representar o sistema através de ações genéricas, a ser manipuladas pelo algoritmo, utilizadas no plano para satisfazer as condições do mesmo.

Pretendeu-se com esta abordagem implementar um sistema com avatares autónomos capazes de desempenhar papéis de membros de uma equipa. A concretização plena deste objetivo permitiria dispor de um ambiente de treino individual que poderia ser utilizado de forma flexível por uma equipa de formandos completa ou incompleta (ou seja, com menos pessoas do que as necessárias à execução da tarefa, por apenas uma pessoa). Poderia mesmo permitir a execução só por avatares autónomos, sem participação de qualquer pessoa, apenas para observação da tarefa em execução automática.

IV. ALGUMAS ABORDAGENS DE INTELIGÊNCIA ARTIFICIAL

Um conjunto de tarefas práticas necessita de uma representação do seu conhecimento para ser interpretada por técnicas de planeamento. Nalgumas técnicas de inteligência artificial usadas habitualmente para planeamento automatizado de ações, essa representação é composta por estados, objetivos e ações. Os estados são expressões básicas que descrevem características iniciais do mundo analisado (ex.: Livre(B) – O objeto “B” encontra-se no estado “Livre”); os objetivos são expressões que representam estados desejáveis que o agente procura atingir; as ações são as operações que o agente pode executar para alterar o estado do mundo. Por sua vez, cada ação tem de ter associado um conjunto de pré-condições (se não estão cumpridas, não é possível executar a ação) e pós-condições (que representam as alterações no estado efectuadas após a execução da ação).

De entre as várias linguagens de representação existentes, foi na linguagem de representação STRIPS (*Stanford Research Institute Problem Solver*) que se baseou este processo [5]. Tratou-se de uma opção meramente de conveniência, face à característica exploratória deste trabalho, por via da familiaridade do implementador com essa representação. Numa abordagem mais orientada para a implementação final, seria de considerar formas mais recentes, como a PDDL [6]. Depois de um sistema estar definido na linguagem STRIPS torna-se mais fácil a interpretação do mesmo por algoritmos. No entanto, a escolha do algoritmo adequado terá que ser com base numa análise específica a cada problema, pois cada algoritmo tem as suas características próprias. Existem diversos algoritmos, mas na componente de ordens e planeamento de tarefas, são

comuns dois tipos: *partial-order planning* e *total-order planning*.

Os algoritmos baseados em *partial-order planning* são algoritmos que constroem um plano sem uma ordem exata de execução. São caracterizados como sendo algoritmos não determinísticos pois existem diversas formas e caminhos para se conseguir atingir o objetivo final, não sendo obrigatório haver um caminho específico. Um plano gerado por estes algoritmos é composto por quatro componentes: um conjunto de ações, uma ordem parcial para as ações, um conjunto de relações causais, um conjunto de condições em aberto, isto é, condições que ainda não foram cumpridas por nenhuma ação do plano parcial. Estas quatro componentes são tratadas durante o processo de geração do plano parcial, processo este que se inicia no estado inicial do sistema e termina quando não existem mais condições em aberto e o objetivo final foi atingido. Este tipo de algoritmos, em cada iteração, identificam uma ação com pré-condição não satisfeita, introduz no plano a ação cuja pós-condição satisfaz a pré-condição não satisfeita, atualiza as relações causais e verifica se existem lacunas no plano, corrigindo o mesmo em caso de problemas [3].

Os algoritmos baseados em *total-order planning* são semelhantes aos anteriores, pois têm a mesma estrutura também visam atingir uma determinada meta. No entanto, geram um plano que consiste num conjunto de ações sequenciais, com uma ordem específica de execução. A figura 6 mostra exemplos de planos gerados por algoritmos de ambos os tipos.

A escolha do tipo de algoritmo depende muito das características do sistema ao qual o mesmo se destina. No caso em análise, o sistema-alvo foca-se em processos executados em equipa, compostos por diversas ações (com pré e pós-condições associadas), podendo cada uma delas ser efetuada por um elemento não específico da equipa. Em particular, em vários momentos a ordem de execução de tarefas pode ser indiferente (como nos casos apresentados de remoção de parafusos ou instalação de apoios). Com base nestas características, um algoritmo baseado em *partial-order planning* é o mais adequado entre estes dois tipos.

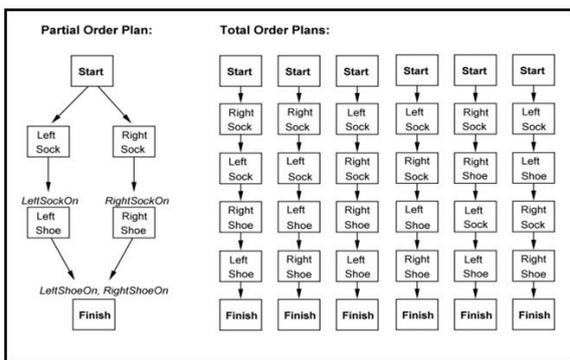


Figura 6 – Planos de ordens parciais e totais [2].

O algoritmo desenvolvido sofreu algumas adaptações para gerar um plano tendo em conta os atores a atribuir a cada ação. Em particular, a adição de um novo campo em cada ação do plano, identificando qual o mecânico ou mecânicos que desempenham determinada operação.

V. AMBIENTE DE TREINO 3D

A análise de tarefas exposta na secção III é independente das características do ambiente concreto de execução. No caso do ambiente físico, não atende a aspetos em que local se encontram as ferramentas ou peças a utilizar nas ações. No caso do espaço de simulação 3D, além destes aspetos de pormenor é ainda preciso atender às formas de interação que a plataforma 3D específica proporciona. Desta forma, foi preciso adaptar as ações gerais a estes condicionalismos de contexto, convertendo cada ação do plano em ações no espaço 3D.

Por exemplo, a ação de remover um parafuso traduziu-se em quatro ações no espaço 3D: chegar junto à fuselagem do avião, fazer *take* do objeto que representa o parafuso, deslocar-se para junto da mesa de ferramentas e fazer *rez* (criar uma representação visual) desse mesmo parafuso em cima da mesa de ferramentas. Estas quatro ações são ilustradas nas figuras 7, 8, 9 e 10, respetivamente.



Figura 7 – Ação “chegar junto à fuselagem do avião”.

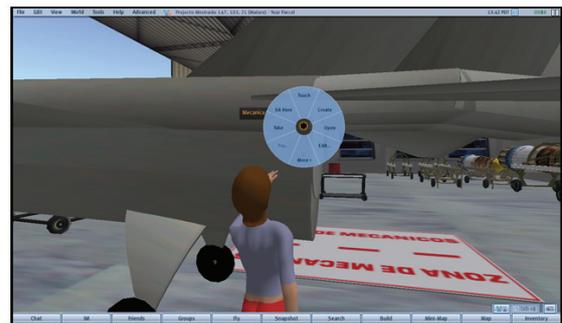


Figura 8 – Ação “fazer *take* do objeto parafuso”.



Figura 9 – Ação “deslocar-se para junto da mesa de ferramentas”.

Este processo de conversão para adaptação ao espaço 3D foi repetido para todas as ações do plano. Foi utilizada uma cópia do espaço 3D desenvolvido no âmbito do projeto já referido de colaboração entre a UTAD e a FAP, que representa um hangar semelhante ao utilizado pelos mecânicos para efetuar as tarefas em ambiente real. De seguida apresentam-se alguns aspetos deste espaço 3D (figuras 11-13).

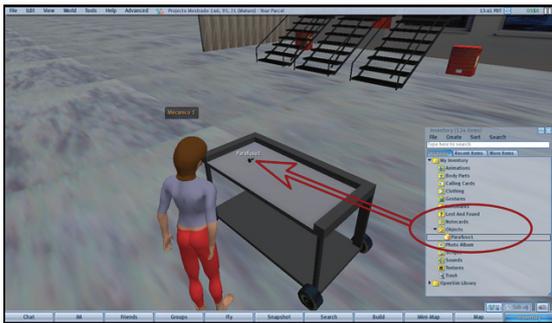


Figura 10 – Ação “fazer *rez* do parafuso em cima da mesa”.

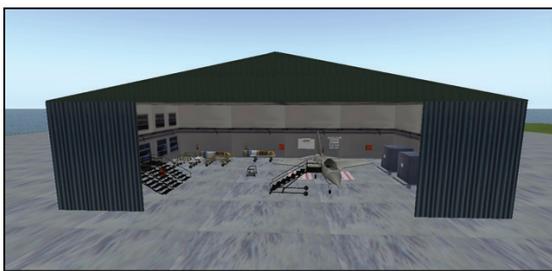


Figura 11 – Hangar para formação de mecânicos.



Figura 12 – Avião F-16 disponível para formação.

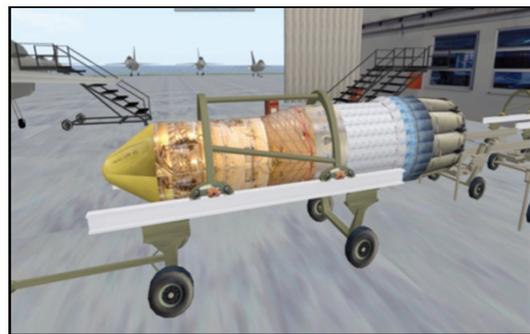


Figura 13 – Motor Pratt & Whitney F100 para formação.

VI. FUNCIONAMENTO DO SISTEMA IMPLEMENTADO

Esta secção apresenta, de um modo geral, o funcionamento do protótipo implementado. Após a ordem de início do processo de treino no espaço 3D, o sistema gera um plano de ações resultante do processamento do algoritmo e guarda esses mesmos dados de forma a poder consultá-los durante o processo. Posteriormente, os objetos 3D do mundo virtual OpenSimulator comunicam via pedidos HTTP com um servidor de *web services* de forma a manter sincronismo e registo de estados entre si (abordagem inspirada na de Fonseca *et al.* [3]). No início do processo de treino, um utilizador indica ao sistema quantos avatares autónomos são necessários, sendo que este sistema foi implementado atendendo a um número máximo de 4 avatares para o treino, no entanto o sistema é escalável e pode ser facilmente adequado para um maior número de avatares autónomos. Os avatares solicitados são consequentemente iniciados pelo sistema, para integrem a equipa. A presença de avatares autónomos durante o processo de treino, leva à necessidade do sistema calcular e atribuir o seu comportamento, de forma dinâmica. Este processo é realizado através do algoritmo de *partial-order planning*, já descrito, que determina os comportamentos necessários para cada avatar autónomo. Os comportamentos são implementados pelo sistema através da biblioteca *libopenmetaverse*, que permite controlar as ações

de avatares, para comunicar e interagir com o espaço virtual – uma abordagem baseada na utilizada por Lopes *et al.* [7].

VII. CONSIDERAÇÕES FINAIS E TRABALHO FUTURO

O objectivo principal deste trabalho foi desenvolver um protótipo que permitisse refletir sobre o recurso a avatares autónomos (agentes inteligentes) para desempenhar o papel de membros na execução de tarefas em equipa. Pretende-se encontrar uma abordagem que permita resolver questões relacionadas com limitações logísticas ao treino de tarefas em equipa, como mais-valia para a formação profissional.

Neste momento, o protótipo está funcional e pode ser testado por mecânicos reais, carecendo contudo de implementação da sequência completa de ações para o efeito (instalação do motor F100 numa aeronave F16). Consideramos que os testes no terreno com equipas mistas de avatares controlados por mecânicos humanos e avatares autónomos controlados por um sistema de decisão será um passo futuro importante para recolha de dados que permitam estudar a validade desta ideia.

REFERÊNCIAS

- [1] Lockheed Martin Corp. (2009). *TO 1F-16AM-2-70JG-10-21 - Organizational Maintenance - Engine Removal and Installation – Model F100-PW-220/220E – USAF/EPAF Series – F-16^B Mid-Life Update aircraft*, Technical Manual Job Guide. Bethesda, MD, EUA: Lockheed Martin Corporation.
- [2] Russel, Stuart; Norvig, Peter (2010, Eds.). *Artificial Intelligence: A Modern Approach*. Third edition, section 11.3, “Partial-Order Planning”, pp. 387-395. Upper Saddle River, NJ, EUA: Pearson Education.
- [3] Fonseca, Benjamim; Paredes, Hugo; Rafael, Jorge; Morgado, Leonel; Martins, Paulo (2011). A Software Architecture for Collaborative Training in Virtual Worlds: F-16 airplane engine maintenance. In Adriana S. Vivacqua, Carl Gutwin and Marcos R.S. Borges (Eds.) "Collaboration and Technology: 17th International Conference, CRIWG 2011, Paraty, Brazil, October 2-7, 2011. Proceedings", 102-109. Berlin: Springer.
- [4] Teixeira, Libânia; Pinto, Isabel (2010). *Guia de instalação de motor em aeronave F-16*. Relatório de Projecto 2009-2010, Licenciatura em Tecnologias de Informação e Comunicação. Vila Real, Portugal: UTAD.
- [5] Fikes, Richard (1993). STRIPS, A Retrospective. *Artificial Intelligence*, 59 (1-2), 227-232.
- [6] Ghallab, M., Isi, C. K., Penberthy, S., Smith, D. E., Sun, Y., and Weld, D. (1998). *PDDL - the planning domain definition language*. Technical report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.
- [7] Lopes, A., Pires, B., Cardoso, M., Santos, A., Peixinho, F., Sequeira, P., & Morgado, L. (2008). Sistema de criação de movimentos de Andebol em Second Life para Formação de Treinadores. *Prisma* com 6, ISSN 1646-3153. 33-49.