

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CIÊNCIA DA COMPUTAÇÃO**

**USO DE STEERING BEHAVIOR PARA SIMULAÇÃO DE MANOBRAS
MILITARES**

TRABALHO DE GRADUAÇÃO

Matheus Ronssani de Figueiredo

**Santa Maria, RS, Brasil
2015**

**USO DE STEERING BEHAVIOR PARA SIMULAÇÃO DE MANOBRAS
MILITARES.**

Trabalho de Graduação a ser apresentado ao curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharel em Ciência da Computação**

Matheus Ronssani de Figueiredo

Orientador: Prof. Dr. Mateus Beck Rutzig

Trabalho de Graduação Nº 402
Santa Maria, RS, Brasil
2015

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

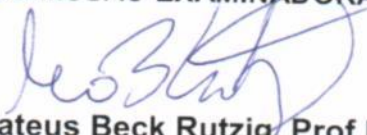
A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

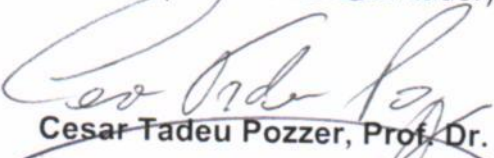
**USO DE STEERING BEHAVIOR PARA SIMULAÇÃO DE
MANOBRAS MILITARES.**


elaborado por
Matheus Ronssani de Figueiredo

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:


Mateus Beck Rutzig, Prof. Dr.
(Presidente/Orientador)


Cesar Tadeu Pozzer, Prof. Dr. (UFSM)


Luis Alvaro de Lima Silva, Prof. Dr. (UFSM)

Santa Maria, 09 de Dezembro de 2015.

RESUMO

USO DE STEERING BEHAVIOR PARA SIMULAÇÃO DE MANOBRAS MILITARES

Autor: Matheus Ronssani de Figueiredo

Orientador: Mateus Beck Rutzig

Este trabalho apresenta um estudo sobre steering behaviors juntamente com uma aplicação desses algoritmos em um projeto de simulação para treinamento militar. Por meio deste, procura-se apresentar uma solução para aumentar a robustez do sistema em relação à falhas dos agentes envolvidos na simulação.

Palavras-Chave: Steering Behavior, Simulação, Treinamento Militar.

SUMÁRIO

1	Introdução	7
2	Steering Behavior	9
2.1	Steering Behaviors Individuais	10
2.1.1	Seek (Buscar)	10
2.1.2	Flee (Fugir)	10
2.1.3	Pursuit (Perseguir)	11
2.1.4	Evasion(Evasão)	11
2.1.5	Arrive(Chegada)	12
2.1.6	Obstacle Avoidance (Evitar Obstáculos)	12
2.2	Steering Behaviors de Grupo	13
2.2.1	Separation(Separação)	13
2.2.2	Cohesion (Coesão)	14
2.2.3	Alignment (Alinhamento)	15
2.3	Combinação de Comportamentos	15
3	Reconhecimento, Escolha e Ocupação de Posição	16
3.1	Recebimento das Ordens.	17
3.2	Trabalhos preparatórios.	17
3.3	Reconhecimento no escalão Grupo (Reconhecimento de 1º Escalão).	17
3.4	Apresentação do relatório e decisão do comandante	18
3.5	Reconhecimento das Baterias (2º e 3º escalões).	18
3.6	Ocupação e Desdobramento do GMF.	18
3.7	REOP na Posição de Tiro.	19
3.7.1	Reconhecimento da Posição de Tiro.	19
3.7.2	Ocupação da Posição de Tiro	20
4	Trabalhos Relacionados	21
4.1	Steering Behaviors for Autonomous Vehicles in Virtual Environments	21
4.2	Behavior-based Formation Control for Multi-robot Teams	21
5	Proposta	24
5.1	Reconhecimento	24
5.1.1	Chegada do marruá à posição de tiro.	25
5.1.2	Chefes de Peça caminhando em formação até a posição da LMU base.	25
5.1.3	Chefes de Peça dispersando da LMU base para posição de sua LMU.	25

5.1.4	Chefes de Peça retornando a posição da LMU base.....	25
5.1.5	Formando posição de grupo dos Chefes de Peça.....	25
5.1.6	Chefes de Peça retornando para o marruá.	25
5.1.7	Marruá saindo da posição de tiro.	25
5.2	Ocupação.....	26
5.2.1	LMUS chegando na posição de tiro.....	26
5.2.2	LMUS direcionando-se para suas respectivas posições.	26
5.2.3	LMUS retirando-se da posição de tiro.	26
5.3	Metodologia	26
6	Implementação.....	28
6.1	Modelos e Controladores.....	28
6.1.1	Veículos	28
6.1.2	Soldados	29
6.2	Steering Behaviors Individuais	30
6.2.1	Arrive	30
6.2.2	Obstacle Avoidance.....	31
6.3	Steering Behaviors de Grupo	32
6.3.1	Reconhecimento de Vizinhos	33
6.3.2	Cohesion	33
6.3.3	Separation	34
6.3.4	Alignment	34
6.4	Desafios Encontrados.....	35
7	Resultados	37
8	Conclusão e Trabalhos Futuros	39
9	Referências	40

1 Introdução

O desenvolvimento da computação nas últimas décadas e sua expansão para diversas áreas como forma de aprimorar serviços ou ferramenta de apoio deu origem a uma transformação acelerada de costumes, práticas e visão do mundo. Segundo [Cunha, 2011] esse avanço deve-se ao exército e a indústria de defesa e é gerado pela necessidade destes possuírem um nível tecnológico superior em relação aos outros exércitos com o fim de manter soberania.

Grande parte das forças armadas no mundo faz uso da tecnologia para simular missões, operações ou ações conjuntas de suas diversas frações (marinha, aeronáutica, forças armadas). Isto ocorre porque é desejável que esses profissionais tenham um alto nível de treinamento sem estarem sujeitos a custos humanos, econômicos e sociais [Garcia, 2006]. O Brasil já faz uso de ferramentas terceirizadas para simulações, um exemplo é o simulador COMBATER, serviço contratado da empresa MASA. Este simulador gerou e continua gerando altos custos para o exército brasileiro, principalmente devido a sua manutenção.

Visando reduzir os custos, evitar desgastes de viaturas e obter o desenvolver tecnologia nacional, o projeto SiS ASTROS 2020 tem como um de seus objetivos desenvolver um simulador virtual tático para o treinamento da doutrina de Reconhecimento Escolha e Ocupação de Posição (REOP) de Baterias ASTROS. O REOP é uma doutrina tática utilizada para missões com Baterias ASTROS. Essa doutrina consiste em duas fases: Reconhecimento e Ocupação. O Reconhecimento é realizado por oficiais do exército e tem como objetivo escolher, demarcar posições de interesse para que a Bateria ASTROS aguarde e realize a missão. Na fase de Ocupação as viaturas e os militares deslocam-se para as posições reconhecidas desdobrando-se nos locais que foram previamente demarcados. As ações de reconhecimento e ocupação são realizadas para duas posições: Espera e Tiro. Na posição de espera, os oficiais e as viaturas aguardam o comando para se deslocar até a Posição de Tiro e cumprir a missão. O requisito principal desta posição é que as viaturas estejam camufladas. Na Posição de Tiro as viaturas ASTROS posicionam-se nos lugares demarcados no reconhecimento, efetuam o tiro e retornam para a Posição de Espera.

Um simulador é visto como um meio de "dar vida" a modelos e situações. A simulação é comumente composta por agentes reais, e então pode ser entendida como uma representação do mundo real [Crookal,1987]. Tendo isso em vista, procurar manter o máximo de realismo em um simulador utilizado para treinamento é essencial, tanto na representação dos agentes como em suas decisões e consequências, com o fim de tornar o resultado do treinamento mais proveitoso. Desta forma, em [Reynolds,1999] foram propostos algoritmos de steering behavior. Estes algoritmos permitem que um agente ou um grupo de agentes sejam reativos ao ambiente. Ou seja, os agentes possuem a capacidade de tomar decisões de forma autônoma, tornando a simulação mais fiel à realidade e economizando

esforços de modelagem em relação a implementação de comportamentos de modo determinísticos.

Este trabalho propõe a utilização de algoritmos de steering behavior no desenvolvimento do Reconhecimento e Ocupação da Posição de Tiro de uma Bateria ASTROS. Assim, o posicionamento das viaturas, movimentos de grupos de oficiais serão realizados de forma autônoma. Com isso, o controle, o movimento, as reações e as ações dos oficiais e das viaturas ASTROS serão mais próximos a disciplina e a conduta militar, refletindo assim em um resultado final mais fidedigno de simulação.

Este trabalho está dividido em 8 capítulos. No capítulo 2 é realizada a contextualização de algoritmos de steering behavior. No capítulo 3 são apresentados os procedimentos de REOP. Alguns trabalhos que utilizam a técnica de steering behavior são discutidos no capítulo 4. O capítulo 5 apresenta a proposta e as tarefas que serão realizadas na implementação dos procedimentos de REOP utilizando algoritmos de steering behavior. No capítulo 6 são discutidos os problemas encontrados, as escolhas feitas e é explicado um pouco sobre a implementação deste trabalho. No capítulo 7 são apresentados os resultados deste projeto. E finalmente no capítulo 8 são discutidos esses resultados, e apresentados trabalhos futuros.

2 Steering Behavior

O termo Behavior, ou comportamento, possui diversos significados, desde várias pequenas ações facilmente previsíveis de um mecanismo simples a uma complexidade imensa de acontecimentos em um sistema caótico. Na computação, é comumente usado como sinônimo de "animação" ou "Inteligência Artificial". [Reynolds,99].

Esses comportamentos são utilizados em agentes autônomos para tornar mais realístico o movimento. Esses agentes são, segundo [Maes, 95], sistemas computacionais inseridos em ambientes complexos e/ou dinâmicos que têm a capacidade de perceber e agir de modo autônomo para atingir objetivos ou executar tarefas para os quais tenham sido modelados.

O chamado comportamento de um agente pode ser entendido de uma maneira melhor se for dividido em camadas. Em um modelo simples seriam 3 camadas: Seletor de Ações, Comportamento e Movimentação.

- Seletor de Ações: O agente terá algum objetivo, estratégia ou função, que irá ser definido por uma Inteligência Artificial. Baseado nesse objetivo, essa camada irá decidir e aplicar um ou mais comportamentos para realizar esse objetivo. Por exemplo, existem dois pontos, A e B, suficientemente distantes um do outro para que haja uma árvore no meio do caminho entre eles. O agente está no ponto A, e seu objetivo é ir ao ponto B sem colidir com a árvore. Então essa camada irá selecionar um comportamento de ir até o ponto B, e desviar da árvore no trajeto.
- Comportamento (Steering): Com a ação selecionada pela camada de Seleção de Ações, o próximo passo é calcular o movimento. Esse movimento será uma força de direcionamento, ou Steering Force. Que após calculada, é repassada para a camada de movimentação onde será aplicada.
- Movimentação: Essa camada está diretamente ligada ao movimento do agente, ela é responsável por transformar as forças que o comportamento gera para movimentar o agente. No exemplo anterior, supondo que o agente seja uma pessoa caminhando, essa camada iria gerar algo como "pé direito, pé esquerdo, pé direito, um passo para o lado, em frente, pé esquerdo", até o objetivo ser alcançado.

Para a camada de movimentação vamos definir um agente com um modelo simples. O agente é composto por:

- Massa : Escalar;
- Posição : Vetor;
- Velocidade: Vetor;
- Força_Máxima: Escalar;

- Velocidade_Máxima: Escalar;
- Direção : Vetor;

Os parâmetros do agente simplificam alguns fatores que não são relevantes para a apresentação do comportamento, Força_Máxima por exemplo, inclui características como aceleração, que tem fatores limitantes em uma locomoção. A física aplicada no agente é baseada na integração pelo método de Euler. Onde a Steering Force gerada pela camada de comportamento, limitada pela Força_Máxima, será aplicada na massa do agente. Desta forma, uma força (Steering Force) é aplicada pelo comportamento à massa do agente gerando uma aceleração na direção resultante do comportamento. Por fim, essa aceleração é adicionada a velocidade atual produzindo uma nova velocidade, que será limitada pela Velocidade_Máxima.

- Steering Force = truncate (Steering Direction, Força_Máxima)
- Aceleração = Steering Force / Massa
- Velocidade = truncate (Velocidade + Aceleração , Velocidade_Máxima)
- Posição = Posição + Velocidade

Com o modelo do agente já definido e um embasamento de como é feita a atualização dos parâmetros do agente, é possível descrever alguns behaviors apresentados por [Reynolds,99].

2.1 Steering Behaviors Individuais

Nesta seção serão descritos comportamentos aplicados a um único agente.

2.1.1 Seek (Buscar)

Esse comportamento direciona o agente em direção a um ponto específico no espaço. O comportamento ajusta a velocidade do agente para direcioná-lo para o ponto selecionado. Como pode ser observado na Figura 1, o Seek gera uma força (representado pela seta azul (Seek steering)) no agente (representado pelo triângulo verde) e modifica a sua velocidade atual (representada pela seta verde(Current Velocity)) fazendo que ele siga pelo caminho traçado em azul (Seek path) até o alvo (Target). Este comportamento deve ser combinado com o Arrive (Seção 2.1.1.5) visto que a utilização do mesmo de forma singular faz com que o agente fique sempre girando em volta do Target.

2.1.2 Flee (Fugir)

O comportamento Flee direciona o agente para a sentido contrária ao alvo sendo oposto do Seek. Como pode ser observado na Figura 1, o Flee gera uma força (representado pela seta vermelha (Flee steering)) no agente (representado pelo triângulo verde) e modifica a sua velocidade atual (representada pela seta verde(Current Velocity)) fazendo que ele siga pelo caminho oposto (Flee path) ao alvo (Target).

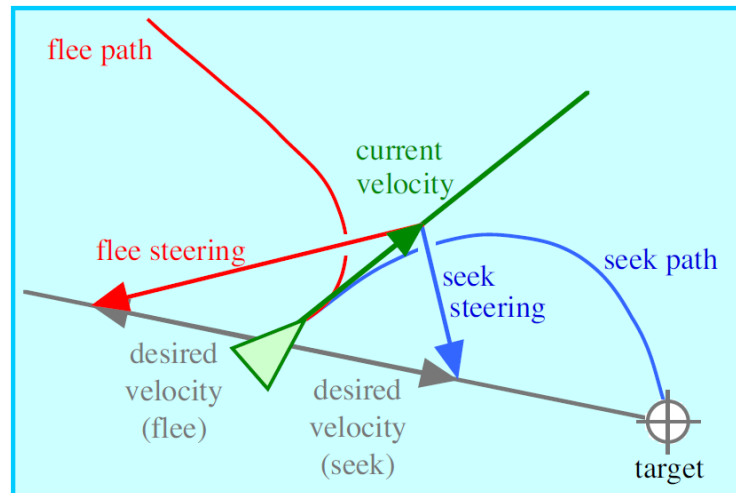


Figura 1 Ilustração do Seek e Flee. Fonte: Reynolds, 2000.

2.1.3 Pursuit (Perseguir)

Pursuit é semelhante ao Seek, diferenciando do último devido ao alvo ser um objeto em movimento. Para esse comportamento é necessário haver uma previsão do movimento do alvo, assim é aplicado o comportamento Seek para a posição prevista do alvo.

Na Figura 2 podemos observar que o alvo (triângulo cinza) está na posição atual (Now) e sem o steering behavior terá um trajeto em linha reta. Assim, o agente com o objetivo de perseguir o alvo faz uma previsão de sua posição no próximo instante (Future) e então utiliza o comportamento Seek levando em conta a posição futura do alvo e gera uma força que fará o agente seguir o trajeto azul (Pursuit).

2.1.4 Evasion(Evasão)

O comportamento Evasion é análogo ao Pursuit, utilizando Flee para evitar a posição prevista do alvo. Então a força gerada pelo comportamento irá resultar no trajeto vermelho (evasion) como pode ser observado na Figura 2.

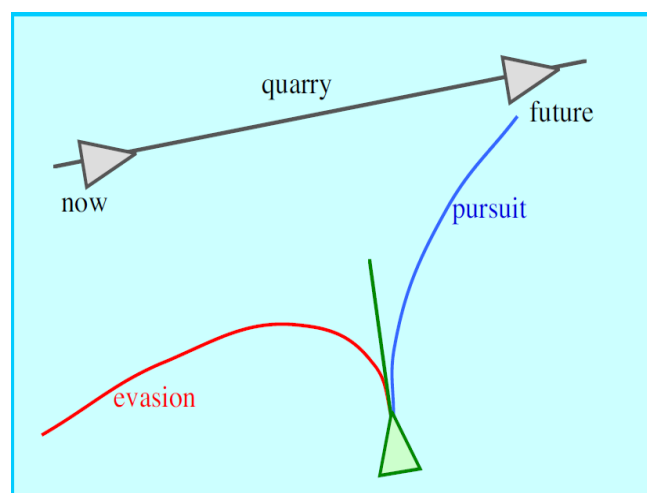


Figura 2 Pursuit e Evasion. Fonte: Reynolds, 2000.

2.1.5 Arrive(Chegada)

O comportamento Arrive assemelha-se ao Seek, ou seja, quando o agente está afastado do alvo, o comportamento gera uma força para que o agente se aproxime do alvo. Diferentemente do Seek, o Arrive, conforme o agente se aproxima do alvo, reduz gradativamente sua velocidade até que a mesma se torne nula quando estiver na mesma posição do alvo. Na Figura 3 podemos observar que o comportamento ao aplicar a força no agente desloca a direção do mesmo em relação ao alvo. Entretanto, a velocidade do agente é reduzida ao ponto que o mesmo se aproxima do alvo.

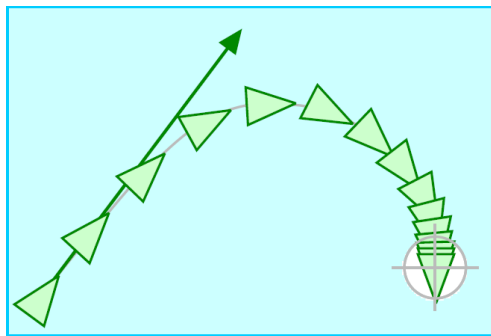


Figura 3 Ilustração do Arrive. Fonte: Reynolds, 2000.

2.1.6 Obstacle Avoidance (Evitar Obstáculos)

O comportamento Obstacle Avoidance faz com que o agente desvie de obstáculos no cenário. Este comportamento usa previsão do movimento do agente para detectar se o mesmo irá colidir com algum objeto. Caso o comportamento detecte que irá ocorrer colisão, o mesmo irá gerar uma força que alterará a direção atual do agente para desviar do objeto.

Uma distinção entre Flee e o Obstacle Avoidance é o fato que este irá desviar apenas se houver uma detecção de uma possível colisão, já o Flee iria desviar do alvo mesmo que não existisse uma colisão, apenas por sua proximidade.

Como pode ser observado na Figura 4, o agente está em um trajeto que irá colidir com o obstáculo B (retângulo azul). Assim, o comportamento gera uma força (seta vermelha) que irá fazer a velocidade do agente mudar de direção. A resultante ficará entre A e C, fazendo que o agente mude o trajetória e evite uma colisão com o obstáculo B.

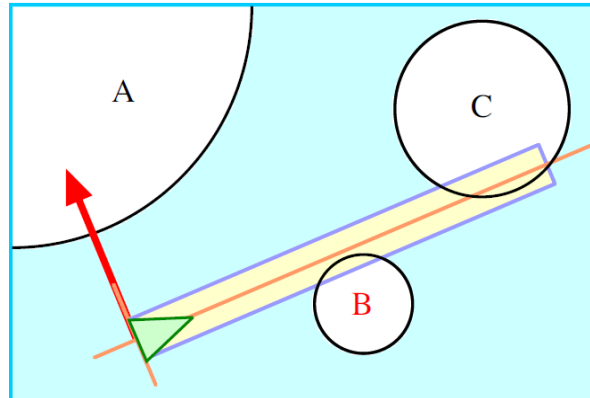


Figura 4 Obstacle Avoidance. Fonte: Reynolds, 2000.

2.2 Steering Behaviors de Grupo

São comportamentos que se aplicam a um agente e influenciam o comportamento de agentes próximos. É feita uma separação de agentes locais para evitar um grande custo computacional para busca de todos os agentes do ambiente, juntamente como o custo para calcular todas as forças resultantes de todos os agentes de um ambiente virtual. Os agentes locais são reconhecidos através de um raio em torno do agente principal. Como ilustrado na Figura 5, onde os agentes que estão dentro desse raio são considerados agentes locais, então os comportamentos que utilizam o grupo para calcular a força resultante, recebem essa informação do radar. Também é considerando um ângulo, que representa o FOV (Field Of View), ou campo de visão do agente, para localizar os agentes vistos pelo agente principal.

Abaixo são apresentados os principais comportamentos de grupos propostos por [Reynolds,99].

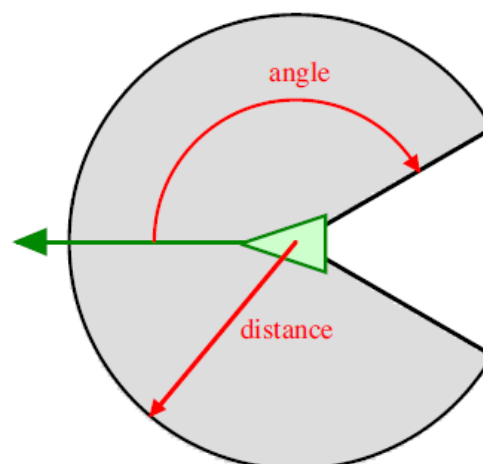


Figura 5 Localização de Agentes próximos. Fonte: Reynolds, 2000.

2.2.1 Separation(Separação)

O comportamento Separation é utilizado para manter certa distância entre os agentes próximos. Esta distância é gerada através de uma força repulsiva entre os agentes evitando que se aglomerem em um único ponto.

Esse comportamento calcula uma força repulsiva para cada agente próximo, então realiza uma soma e uma normalização das mesmas em função do raio de localização dos agentes, resultando na força final.

Na Figura 6 pode-se ver que o agente principal (verde) possui 3 agentes vizinhos dentro do seu raio de localização. Nessa caso, a força resultante, vermelha, aplica-se para traz em direção a direita, já que a maior força de repulsão foi a do agente mais próximo (o superior esquerdo).

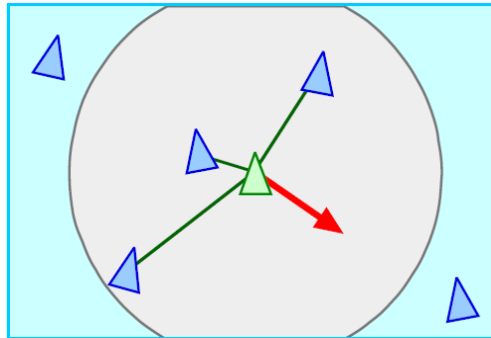


Figura 6 Separation. Fonte: Reynolds, 2000.

2.2.2 Cohesion (Coesão)

O Cohesion permite aos agentes agruparem-se ao redor de um ponto médio que é calculado a partir das posições dos agentes que estão próximos. Então a força resultante desse comportamento é aplicada em direção a esse ponto médio para fazer o agente aproximar-se do grupo e manter sua posição. Na figura 7 podemos notar que este ponto médio, demarcado com um círculo verde, e todos os agentes próximos são "conectados" a ele através da força gerada pelo comportamento. Assim, o agente principal (verde) exerce uma força resultante (vermelha) direcionada para o ponto médio do grupo.

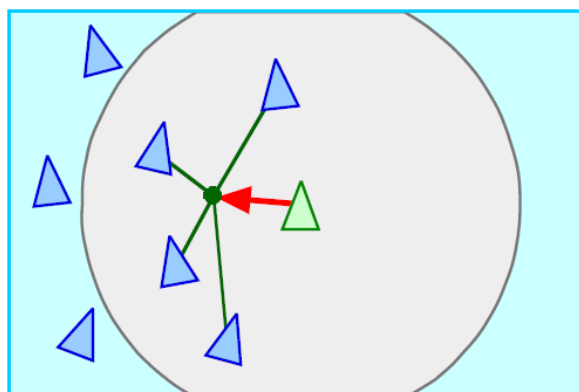


Figura 7 Cohesion. Fonte: Reynolds, 2000.

2.2.3 Alignment (Alinhamento)

O comportamento Alignment permite ao agente alinhar-se com os outros agentes. Isso é feito pelo cálculo de uma média entre Direção e Velocidade de todos os agentes próximos. Na figura 8 é mostrado uma situação onde o agente principal (verde) está usando o comportamento Alignment para corrigir sua Direção em relação aos agentes próximos. As direções são representadas pelas linhas à frente do agente. Então a força resultante do comportamento irá corrigir a direção do agente mantendo a mesma velocidade.

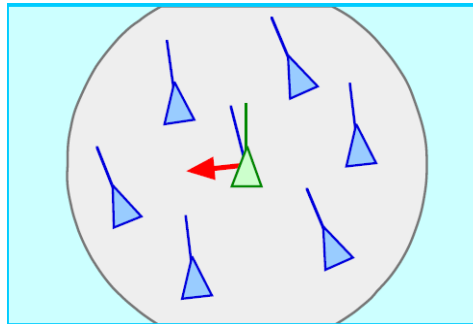


Figura 8 Alignment. Fonte: Reynolds, 2000.

2.3 Combinação de Comportamentos

Os comportamentos apresentados anteriormente são utilizados para compor padrões mais complexos de comportamentos. Para se obter um comportamento mais real são selecionados e aplicados vários comportamentos em um único agente.

Essa combinação pode ocorrer de duas maneiras, o agente pode sequencialmente ir aplicando vários comportamentos ou executá-los em paralelo.

A primeira aproximação é feita pela camada de Seleção de Ação, que irá repassar para o agente a alteração do comportamento. Como por exemplo um agente pode estar utilizando um Flee de um agente que o estava perseguindo, mas aproximou-se de um ponto que era seu objetivo, então ele troca seu comportamento para um Arrive. Porém esse método gera algumas incoerências, pois no fim o agente iria deixar de ser alcançado pelo agente que o estava perseguindo para parar em um objetivo.

A outra maneira seria executar os comportamentos em paralelo, atribuindo pesos para suas forças resultantes, gerando uma única força resultante que iria ser passada para a camada de movimentação. Essa maneira é feita na camada de comportamentos, onde os comportamentos ativos são repassados pela camada de seleção e os pesos atribuídos.

Um exemplo é um agente que está fugindo de outro agente com o Flee, porém depara-se com um obstáculo a sua frente, então é ativado o comportamento de Obstacle Avoidance e um peso atribuído a ele, então a força resultante seria suficiente para fazer o agente continuar sua fuga e evitar a colisão no obstáculo.

3 Reconhecimento, Escolha e Ocupação de Posição

O reconhecimento, escolha e ocupação de Posição (REOP) é um conjunto de ações táticas que permitem o emprego de viaturas ASTROS em uma missão fornecida pelo Estado Maior do Exército. A finalidade do REOP é possibilitar um conjunto de operações a serem executadas com a finalidade de deslocar de uma zona de reunião para uma área de posição, a fim de que o apoio de fogo possa ser iniciado. Sua execução judiciosa permitirá obter volume e potência de fogo, bem como a rapidez no engajamento do inimigo, permitindo que o escalão apoiado possa contar com o apoio de fogo no momento e locais necessários à sua manobra.

Zona de reunião (ZReu) é a área onde encontra-se o alto escalão no início da missão, juntamente com as viaturas e os oficiais do exército. Área de posição (Área Pos) é definida como a parte do terreno ocupada ou a ser ocupada por uma unidade de artilharia. Ela é formada por Posições de Tiro (Pos Tiro) e Posições de Espera (Pos Espa). Cada área de posição é constituída de uma Pos Espa e quatro Pos Tiro, porém essa quantidade pode ser alterada de acordo com o estudo da missão.

Cada uma dessas áreas tem algumas características em relação a sua dimensão e condições que deve apresentar.

- Área de Pos: Deve ter em torno de 9 km de largura e 3 km de comprimento;
- Pos Espa: Não possui um tamanho e formato definido, porém deve apresentar uma boa capacidade de camuflar as viaturas, instalações e oficiais;
- Pos Tiro: Possui 1 km de extensão e 0,6 km comprimento. Deve ser próxima à Pos Espa. O ideal é entre 5 a 10 minutos de deslocamento entre elas. Uma posição de tiro deve ter 1 km de distância de outra posição de tiro
- Área de Trens: os trens do GMF(Grupo de Mísseis e Foguetes) são constituídos pela seção de comando e logística, sendo normal montar postos de distribuição para o suprimento das Baterias (Bia) MF.

Para a seleção das posições, são levados em contas fatores que podem garantir o sucesso de uma missão.

- Deslocamento - Condições de trafegabilidade (estradas, movimento através do campo e dos obstáculos acaso existentes) e segurança para o acesso às áreas de posição selecionadas.
- Circulação - Possibilidades de movimentação no interior da área de posição (natureza do solo e obstáculos acaso existentes) e prováveis efeitos das mudanças nas condições meteorológicas sobre a consistência do terreno.
- Segurança - Avaliação da camuflagem, do espaço para a dispersão, dos obstáculos interpostos entre a área de posição e o inimigo, da

facilidade de ocupação das posições selecionadas, da distância da linha de contato e da proximidade da reserva.

- Coordenação - Necessidade de coordenar a escolha de qualquer posição com unidades vizinhas, escalão superior e outras unidades e instalações da força apoiada.

O REOP é dividido em fases, que podem ocorrer simultânea ou sucessivamente:

1. recebimento das ordens (verbais ou escritas);
2. trabalhos preparatórios;
3. reconhecimento no escalão grupo(Reconhecimento de 1º Escalão);
4. apresentação dos relatórios;
5. decisão do comandante do GMF(Grupo de Mísseis e Foguetes);
6. reconhecimento das Bia LMF(Reconhecimento de 2º e 3º Escalão);
7. ocupação da posição e desdobramento do GMF.

3.1 Recebimento das Ordens.

Para dar-se início uma missão, o escalão superior envia as Mensagens de Tiro, que podem ser verbais ou escritas, para o comandante do GMF com informações da missão que deve ser efetuada.

3.2 Trabalhos preparatórios.

Após o recebimento das ordens, o comandante do GMF fará um estudo da situação para tomar uma decisão preliminar sobre a missão e repassar ao seus oficiais para darem início ao reconhecimento.

O estudo de situação de um comandante pode assumir aspectos diferentes, conforme o tipo e a missão e, particularmente, conforme a ocasião em que é feito, exigindo, algumas vezes, uma adaptação para atender a casos particulares. Caberá ao comandante propor o processo a ser utilizado no engajamento dos alvos, de acordo com os fatores da decisão: Missão, Inimigo, Terreno, Meios e Tempo Disponível.

Após tomada a decisão é formulada a ordem preparatória (a quem apoiar e horários para realização das atividades) e o plano de reconhecimento (composição e missão, transporte, data e hora das reuniões, região a reconhecer).

3.3 Reconhecimento no escalão Grupo (Reconhecimento de 1º Escalão).

Cada oficial designado pelo comandante executa o reconhecimento detalhado, levando em consideração as condições necessárias à ocupação da posição pelos diferentes órgãos do GMF. Neste reconhecimento, são realizados os trabalhos que servirão como base para o reconhecimento das baterias (2º e 3º escalão) como por exemplo:

- Estudo de situação na carta: tem por finalidade selecionar áreas de posição, regiões de Postos de Comando (PC) e itinerários. Nessa

ocasião, também, são realizados estudos preliminares sobre o levantamento topográfico (Lev Topo), instalação das comunicações e outras atividades.

- Plano de reconhecimento: Após o estudo de situação na carta, a decisão preliminar do comandante é consubstanciada no Plano de Reconhecimento. Neste são especificadas: a constituição do reconhecimento, missão aos elementos subordinados, hora e local para apresentação dos relatórios, bem como medidas logísticas que se fizerem necessárias.
- Organização e constituição do reconhecimento:
 - O reconhecimento do GMF é dividido em dois grupos. Um destinado a reconhecer as Áreas de Posição e outro com a finalidade de reconhecer as Áreas para o PC.
 - A constituição das equipes de reconhecimento não é fixa, podendo variar em função da disponibilidade de pessoal, material e, principalmente, do tempo.

3.4 Apresentação do relatório e decisão do comandante

Após o reconhecimento de primeiro escalão, são apresentados os relatórios para o comandante do GMF. A partir dos relatórios apresentados, o comandante decide apoiando ou modificando sua decisão preliminar.

3.5 Reconhecimento das Baterias (2º e 3º escalões).

Após a decisão do comandante, inicia-se imediatamente o reconhecimento de 2º e 3º escalão. O reconhecimento tem por finalidade, verificar mais detalhadamente as características das Áreas de Posição e das Áreas de Trens, tendo em vista levantar subsídios para a escolha das mesmas.

Alguns fatores de seleção para a área de Pos da Bia MF são: camuflagem, espaço para a dispersão, obstáculos, efeitos das condições meteorológicas, natureza do solo para a circulação no interior da posição, amplitude do setor de tiro, alcance, distância do inimigo, controle do espaço aéreo, entre outros.

Com as posições selecionadas, começa o processo de estaqueamento dos locais, quando é marcado o local onde as viaturas da Bateria ASTROS serão alocadas na fase de ocupação.

3.6 Ocupação e Desdobramento do GMF.

Após realizados os reconhecimentos de 1º e 2º Escalão inicia-se a ocupação das posições previamente reconhecidas. Todo grupo deslocar-se-á ao Ponto de Liberação, local onde todo grupo encontra-se antes de ser liberado para ir para suas respectivas posições, onde os elementos subordinados assumem suas funções e deslocam-se em seus itinerários específicos, deslocando-se para suas respectivas áreas de PC ou de Posição.

Ao chegarem em suas posições, o GMF está pronto pra cumprir a Missão de Tiro, que apresenta uma sequência de ações: Mensagem de Tiro, Ordem de Tiro e Comando de Tiro.

- A mensagem de tiro: a fim de abater alvos designados pelo escalão superior ou por outros escalões de artilharia. Esta apresenta as seguintes características: localização do alvo, dimensão do alvo, tipo de foguete a ser empregado, nível de saturação do foguete, entre outros;
- Ordem de tiro: após receber a missão de tiro do escalão superior, é realizada uma breve análise, como: tempo disponível para o tiro, área alvo, foguete empregado, nível de certeza e nível de saturação, próxima posição de tiro, entre outros;
- Comando de tiro: após receber a ordem de tiro, os dados são inseridos no computador de tiro e as elevações iniciais para a regulação das lançadoras de foguetes. Por fim, o tiro é realizado.

Com a Missão de Tiro cumprida, ocorre então a desocupação, onde as viaturas retiram-se das Posições o mais rápido possível para evitar fogo inimigo para revidar o ataque, já que a localização do GMF foi revelada.

3.7 REOP na Posição de Tiro

Nesta seção será apresentado mais detalhadamente o REOP na Posição de Tiro, bem como os oficiais envolvidos no processo.

3.7.1 Reconhecimento da Posição de Tiro

Essa fase tem início no momento em que duas Marruás se deslocam da Posição de Espera até a Posição de Tiro. Para esta tarefa se deslocam o Comandante da Bateria (Cmt Bia), o Comandante de Linha de Fogo (CLF), seis Chefes de Peça (CP) de LMUs, um Chefe de Peça da UCF, um Sargento Auxiliar de Reconhecimento (Sgt Aux Rec) e um Comandante de Seção de Reconhecimento, Comunicações e Meteorologia (Cmt Sec Rec Com Meteo).

Ao chegarem na Posição de Tiro, o Cmt Bia, o CLF e os sete CPs desembarcam das Marruás. Neste local, por ordem do Cmt Bia, o CLF determina a posição da UCF e o CP da UCF estaqueia esta posição que deve ser perpendicular a Direção Geral de Tiro (DGT).

Após, uma das Marruás segue para o Ponto Afastado, com o Cmt Sec Rec Com Meteo. No local, o Cmt Sec Rec Com Meteo marca com uma bandeirola um ponto visível a todas as viaturas da bateria.

Ao mesmo tempo, os seis CP das LMUs irão até o local determinado pelo CLF onde ficarão as lançadoras. O CP da LMU base fica na posição, e os outros 5 CP dispersam-se lateralmente entre 100 e 200 metros, conforme às ordens do CLF.

Quando todos os oficiais envolvidos possuírem visão da posição de todas as outras viaturas na posição de tiro, o CLF dá ordem de estaqueamento. Então todos os CP, o Cmt Sec Rec Com Meteo e o Sgt Aux Rec utilizam seus GPS para retirar as coordenadas de seus respectivos pontos.

Com as coordenadas salvas, todos retornam para a posição da UCF. Assim, o CLF demarca a posição de entrada e saída da bateria da Posição de Tiro, delimitando o itinerário e a ordem de movimento. Por fim, todos embarcam nas duas Marruás e retornam à Posição de Espera.

3.7.2 Ocupação da Posição de Tiro

Nessa fase uma bateria, composta por 6 LMUs, 1 UCF e 1 PCC, sai da Posição de Espera e chega na Posição de Tiro. Então cada viatura, seguindo uma única rota, ocupa sua respectiva posição demarcada no reconhecimento.

As LMUs seguem em fila até as posições estaqueadas, onde então o CP de cada LMU desembarca para balizá-la até a posição exata a ser ocupada. Assim, inicia-se o processo de nivelamento e pontaria, onde a plataforma de tiro é elevada e direcionada para a Direção Geral de Tiro (DGT).

A UCF segue para sua posição, onde é balizada de maneira perpendicular a DGT. O radar é elevado e direcionado para as coordenadas das LMUs e do Ponto Afastado. Calcula os dados de correção do vento e repassa, via rádio, para as LMUs.

Com os dados inseridos no sistema das LMUs, é dado início a decontagem e o disparo é efetuado. Logo após é dada a ordem de auto cessar fogo e mudança de posição, as LMUs baixam a plataforma de tiro e a UCF baixa o radar. Assim, todas as viaturas iniciam o desnivelamento e em seguida retornam a Posição de Espera, sem respeitar a ordem de entrada, pela mesma rota, evitando colisões entre elas.

4 Trabalhos Relacionados

Para o embasamento deste trabalho foi realizada uma pesquisa bibliográfica, a fim de obter-se trabalhos com propostas semelhantes a este. Os resultados desta pesquisa serão discutidos a seguir.

4.1 Steering Behaviors for Autonomous Vehicles in Virtual Environments

Nesse trabalho, [Wang et al., 2005] apresentam uma solução utilizando steering behaviors para simulação de veículos autônomos em ambientes virtuais. Hank simula ambientes virtuais urbanos com veículos circulando por toda rede viária. As estradas são modeladas como faixas contínuas de pontos, interligadas nas extremidades por pontos comuns aos segmentos, permitindo assim o uso de coordenadas naturais pelo framework, o que facilita o cálculo de distâncias entre intersecções e veículos.

Os veículos possuem apenas características simples para o controle dos steering behaviors, aceleração e ângulo de direção. As limitações de movimento do veículo são descritas em duas características: a aceleração serve como força máxima aplicada no veículo; e o ângulo limitante das rodas a rotação máxima do veículo, impedindo curvas bruscas.

No projeto, foram utilizadas duas classes de comportamentos: aceleração e direção. Os comportamentos de aceleração utilizados foram definidos para ultrapassagem, controle de intersecções e tráfego em vias, responsáveis pelo controle da velocidade do veículo em cada caso. A classe de direção é composta pelos comportamentos de mudança de pista e manter a rota numa pista.

Cada um desses comportamentos é independente do outro, não exigindo um controle complexo de suas ações. Isso é feito com o framework definindo as rotas ou destinos dos veículos, então os comportamentos controlam o movimento do veículo até seu objetivo.

Os resultados da combinação de suas forças resultantes são aplicados com pesos para o momento. Por exemplo, o comportamento de ultrapassagem possui um peso muito pequeno em um momento que o controle de intersecção informa que há tráfego no cruzamento de duas vias, forçando o veículo a reduzir.

No caso do projeto SiS ASTROS 2020, não existe uma rede viária para o controle do trajeto da viatura, visto que o projeto precisa simular ambientes de rios cerrados e pântanos, onde não há garantia de vias. Mas a ideia de utilização da limitação do veículo, aceleração e ângulo de rotação, para cálculo do veículo pode ser adaptada para o projeto.

4.2 Behavior-based Formation Control for Multi-robot Teams

Nesse trabalho [Balch et al., 1999] apresenta uma aproximação utilizando comportamentos para manter formação dos robôs. O simulador é baseado na arquitetura AuRA (Autonomous Robot Architecture), uma arquitetura baseada em

motor schemas. Os comportamentos foram feitos utilizando UGV(Unmanned Ground Vehicle) Demo II Architecture, sistema que provê características básicas para controladores e comportamentos de agentes.

Os comportamentos são implementados como processos assíncronos e concorrentes, gerando resultados em tempo de execução que devem ser gerenciados o quanto antes para não interferir no resultado final por causa do atraso das aplicações das forças no agente. Isso é feito utilizando o sistema DAMN Arbiter, integrado ao UGV, que torna o controle reativo baseado em *motor schemas* mais veloz.

A solução apresentada para os comportamentos de formação do grupo é separada em duas etapas, *detect-neighbors* e *maintain-formation*. Esse foi o método selecionado pois segundo Balch, retorna os resultados em menos tempo em comparação com uma maneira feita em apenas uma etapa.

A primeira etapa, *detect-neighbors*, é responsável por calcular as distâncias de todos os agentes vizinhos, e com as distâncias relativas, conseguir identificar o tipo de formação atual.

A segunda etapa, *maintain-formation*, consiste em cada agente calcular a força necessária para manter ou ajustar a formação, levando em conta obstáculos estáticos no terreno.

São trabalhados 3 tipos de formação no trabalho, *Unity-Center-referenced*, *Leader-referenced* e *Neighbor-referenced*, cada uma delas usa um referencial diferente para a condição de distância e da formação.

A *Unity-Center-referenced* utiliza um cálculo relativo individual, ou seja, cada agente calcula sua distância relativa ao centro da formação. Para isso ela calcula a distância de todos os outros agentes contidos na formação, para então descobrir o ponto médio da formação, e então utilizar esse ponto médio no resultado de sua própria distância relativa.

A *Leader-referenced* faz o cálculo da distância em relação ao líder da formação, no caso foi selecionado o agente com o menor ID. Nesse caso, o líder não preocupa-se em manter a formação, apenas os outros agentes participantes da formação são responsáveis por sua manutenção.

Já na *Neighbor-referenced* todos os agentes fazem o cálculo de formação baseado no cálculo dos vizinhos, mantendo a distância máxima e mínima permitida dos vizinhos.

Os resultados apresentados no projeto demonstram que a aproximação *Neighbor-referenced* tem o melhor desempenho e resultado na formação final dos agentes.

Para o projeto SiS ASTROS 2020, pode-se utilizar a aproximação *Neighbor-referenced*, visto que com o número de LMUs, soldados e Marruás é variável, a escalabilidade da solução é a mais eficiente, já que é usado apenas os vizinhos para os cálculos das distâncias relativa e para o resultado da formação final, resultando em um número menor de cálculos para manter a formação dos agentes.

5 Proposta

Conforme já demonstrado anteriormente, o desenvolvimento do simulador virtual tático para o REOP tem como objetivo treinar oficiais do exército brasileiro em missões táticas trazendo benefícios como redução de custos, evitar desgaste das viaturas, etc.

Visto isso, este trabalho propõe a implementação das atividades apresentadas na seção 2.3 utilizando algoritmos de steering behavior para controlar as ações de movimento de objetos envolvidos na simulação.

A escolha de steering behavior para esse trabalho é suportada através de um conjunto de fatores:

- É uma técnica robusta e tolerante a falhas, evitando que aja algum erro que acarrete na parada de algum dos agentes;
- Facilitam o controle de grupos de agentes;
- E apresentam uma teoria matemática e física bem fundamentada [Reynolds,00].

A seguir são apresentadas as escolhas dos comportamentos para cada fase do REOP na posição de tiro. E em seguida é apresentada a metodologia utilizada para este trabalho.

5.1 Reconhecimento

No momento do reconhecimento da posição de tiro existem sete atividades que podem ser implementadas utilizando algoritmos de steering behavior. Estas atividades e os algoritmos que são utilizados estão explicitadas na Tabela 01.

Tabela 01. Comportamentos selecionados no Reconhecimento.

Momento(Cenário)	Agentes(modelo computacional)	Comportamentos
Chegada do marruá à posição de tiro.	Marruá	<i>Obstacle Avoidance</i> combinado com <i>Arrive</i> .
Chefes de Peça caminhando em formação até a posição da LMU base.	Soldados	Combinação de <i>Cohesion</i> , <i>Alignment</i> e <i>Arrive</i> (aplicado no ponto médio do grupo).
Chefes de Peça dispersando da LMU base para posição de sua LMU.	Soldados	<i>Arrive</i> (posição da LMU)
Chefes de Peça retornando a posição da LMU base.	Soldados	<i>Arrive</i>
Reformando posição de grupo dos Chefes de Peça.	Soldados	<i>Cohesion</i> , <i>Alignment</i>

Chefes de Peça retornando para o marruá.	Soldados	<i>Cohesion, Alignment, Arrive</i> (Aplicado no ponto médio do grupo)
Marruá saindo da posição de tiro.	Marruá	Combinação de <i>Arrive</i> e <i>Obstacle Avoidance</i> .

5.1.1 Chegada do marruá à posição de tiro.

Este cenário consiste na saída do marruá da zona de reunião e sua chegada na posição de tiro. O comportamento *Obstacle Avoidance* foi selecionado para esse agente pois existem obstáculos no itinerário, como árvores, que devem ser desviados. Neste cenário o comportamento *Arrive* também foi escolhido para demarcar o ponto onde o marruá deve estacionar para que os oficiais possam desembarcar.

5.1.2 Chefes de Peça caminhando em formação até a posição da LMU base.

Neste cenário os agentes descem do marruá, ficam em formação para o teste do rádio e se preparam para o encontro das posições onde as LMUs deverão estacionar na fase de ocupação. Desta forma, os militares seguem em formação até o ponto onde será fixada a LMU base. Foram escolhidos os comportamentos *Cohesion* e *Alignment* para manter a formação entre os agentes e o *Arrive* para o momento de encontro do ponto específico de escolha da LMU base.

5.1.3 Chefes de Peça dispersando da LMU base para posição de sua LMU.

Após a chegada à posição da LMU base, cada um dos Chefes de Peça segue para o seu objetivo, marcar a posição da sua LMUs. Então foi escolhido o comportamento *Arrive*, para direcionar cada agente para a sua posição.

5.1.4 Chefes de Peça retornando a posição da LMU base.

Após a fase de marcar as posições de cada LMU, então todos os oficiais retornam para o ponto da LMU base. Nesse processo foi novamente selecionado o *Arrive*, já que é um trajeto até um ponto de encontro.

5.1.5 Formando posição de grupo dos Chefes de Peça.

Quando todos os Chefes de Peça retornam a posição da LMU base é retomada a formação de grupo. Para isso foi escolhido uma combinação de *Cohesion* e *Alignment* para ordenar a formação.

5.1.6 Chefes de Peça retornando para o marruá.

Nessa fase os Chefes de Peça retornam para a posição onde estava o marruá, para então seguirem para a próxima posição de tiro ou para a posição de espera. Portanto, como existe um ponto comum como objetivo, foi escolhido *Cohesion* e *Alignment* para manter a formação e o comportamento *Arrive* para o ponto médio do grupo com uma posição próxima ao marruá.

5.1.7 Marruá saindo da posição de tiro.

Após os oficiais embarcarem no marruá, este então sai da posição de tiro, com destino a outra posição de tiro ou a posição de espera. *Arrive* foi selecionado

para direcionar o marruá para o próximo objetivo, e *Obstacle Avoidance* para que evite colisões com objetos do cenário.

5.2 Ocupação

A Tabela 2 ilustra os algoritmos selecionados para as atividades que compõem a ocupação da posição de tiro.

Tabela 02. Comportamentos selecionados na Ocupação

Momento(Cenário)	Agentes(modelo computacional)	Comportamentos
LMUS chegando na posição de tiro.	LMU	<i>Separation, Cohesion, Obstacle Avoidance e Arrive</i>
LMUS direcionando-se para suas respectivas posições.	LMU	<i>Arrive</i>
LMUS retirando-se da posição de tiro.	LMU	<i>Separation, Arrive e Obstacle Avoidance.</i>

5.2.1 LMUS chegando na posição de tiro.

Nessa etapa as LMUs saem da posição de espera para a posição de tiro. Então foram escolhidos *Separation, Cohesion* e *Alignment* para manter a formação e evitar que as viaturas colidam entre si. Para o trajeto foi escolhido *Arrive* já que as viaturas possuem um itinerário definido até um ponto na posição de tiro, onde então dispersam para suas posições. O comportamento *Obstacle Avoidance* foi selecionado para evitar colisões com objetos do terreno.

5.2.2 LMUS direcionando-se para suas respectivas posições.

Para cada viatura direcionar-se à sua posição foi escolhido *Arrive*. Assim. Cada viatura seguirá para o local onde foram definidas suas posições no momento do reconhecimento.

5.2.3 LMUS retirando-se da posição de tiro.

Após efetuar o tiro, as viaturas devem retirar-se imediatamente da posição de tiro para a posição de espera, então foram selecionados os comportamentos *Arrive* e *Separation* para sua saída da posição. O objetivo do *Arrive* é levar as viaturas até o destino, e o *Separation* para evitar colisões entre as LMUs. Também foi escolhido o *Obstacle Avoidance* para evitar colisões com outros objeto durante a trajetória.

5.3 Metodologia

Primeiramente, a proposta do simulador e das atividades nele contidas foram modeladas através do uso do padrão UML (Unified Modelling Language - Linguagem de Modelagem Unificada) no software Bizage através de um modelo de processos.

Em seguida definiu-se o ambiente Unity 3D para o desenvolvimento da aplicação, na versão 5.2.2f1. A escolha se deu devido ao fato de ser uma engine

gratuita e bem difundida para aplicações gráficas e jogos, também devido ao fato desta engine oferecer vários recursos avançados de computação gráfica e física.

Essa engine apresenta três opções de linguagens, Boo Script, C# e JavaScript, dentre estas foi escolhido C# para a o desenvolvimento.

6 Implementação

Os scripts iniciais utilizados para o projeto foram adaptados de uma biblioteca pertencente ao MIT (sob a licença <http://www.opensource.org/licenses/mit-license.php>, disponível em <https://github.com/ricardojmendez/UnitySteer>). Após feito um estudo sobre a versão atual dos comportamentos, foram então feitas adaptações e modificações para o funcionamento no projeto SiS ASTROS 2020.

A seguir será explicado o funcionamento dos controladores de movimento dos agentes, a implementação dos comportamentos e os problemas encontrados durante essa fase.

6.1 Modelos e Controladores

Os controladores são os algoritmos responsáveis por efetuar o movimento dos modelos, gerenciar as animações de movimento e manter as variáveis de controle do modelo, como aceleração máxima e massa.

6.1.1 Veículos

Para fins de controle, a Marruá e a LMU, correspondentes aos modelos apresentados nas figuras 9 e 10 respectivamente, foram implementados sobre o mesmo controlador.

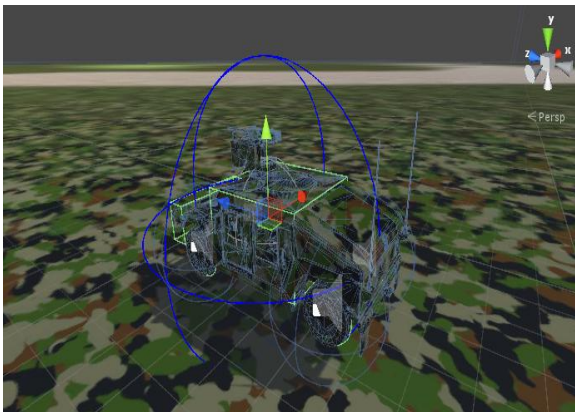


Figura 9 Modelo da Marruá. Fonte: Próprio autor.

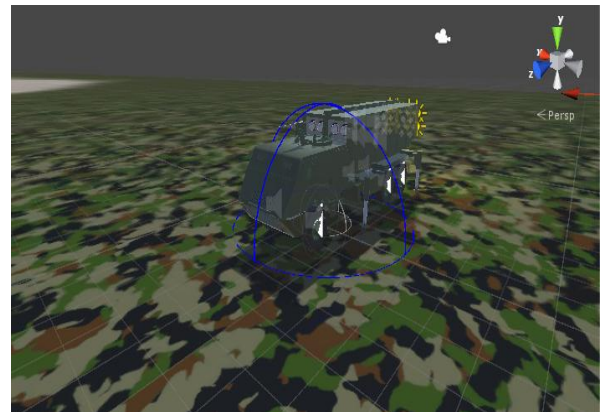


Figura 10 Modelo da LMU. Fonte: Próprio autor.

Pode-se notar na Figura 11 as variáveis pertencentes ao controlador do veículo. Essas variáveis descrevem o veículo de uma maneira computacional, com tamanho (Radius), velocidade mínima para fazer uma curva (Min Speed For Turning), tempo de realizar uma curva (Turn Time), peso (Mass), precisão de chegada (Arrival Radius), velocidade máxima (Max Speed), força máxima que pode ser aplicada no veículo (Max Force), aceleração (Acceleration Rate) e frenagem (Deceleration Rate).

As demais variáveis apresentam funções diversas como restrição de movimento ou gerenciamento dos comportamentos. Allowed Movement Axes permite definir em quais eixos o veículo pode se movimentar. Queue Name, Tick Length e Max Queue Processed fazem a gerencia dos comportamentos, a Queue

Name permite armazenar os comportamentos ativos no agente, Tick Length a

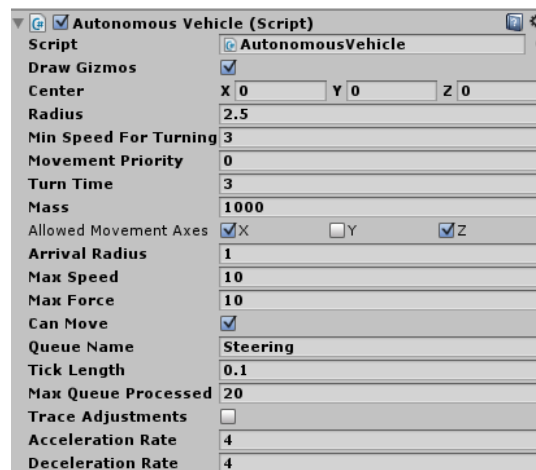


Figura 11 Controlador de Veículos. Fonte: Próprio autor.

quantia de tempo, em segundos, que é feita a atualização do controlador segundo os comportamentos e Max Queue Processed permite armazenar pré-processamentos de alguns comportamentos, como previsões de movimento de alvos.

6.1.2 Soldados

Os soldados foram modelados como ilustrado na Figura 12, um modelo com um movimento bípede, o que impede de usar o mesmo controlador dos veículos pois as variáveis de controle possuem características diferentes, como ângulo máximo de rotação, velocidade mínima de rotação, entre outras.

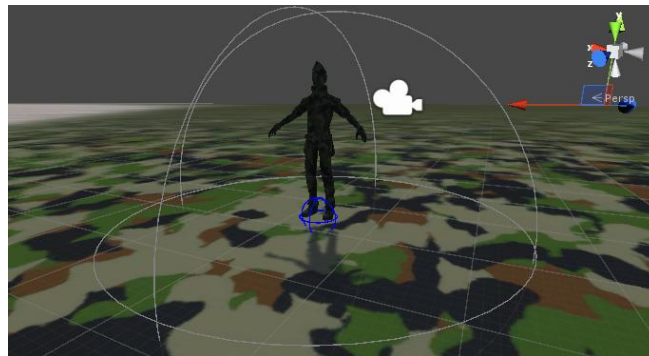


Figura 12 Modelo do Soldado. Fonte: Próprio autor.

Pode-se notar na Figura 13 que muitas variáveis do objeto bípede são as mesmas dos veículos, porém apresentam valores diferentes. A velocidade para fazer uma curva e o tempo de curva são inferiores ao de um veículo já que um humano consegue fazer curvas mais facilmente pois não apresenta limitações de rotação de eixos. A velocidade máxima também foi reduzida juntamente com a precisão de chegada e o peso.

As variáveis de aceleração e frenagem foram retiradas, pois um humano consegue iniciar e finalizar um movimento mais bruscamente que um veículo.

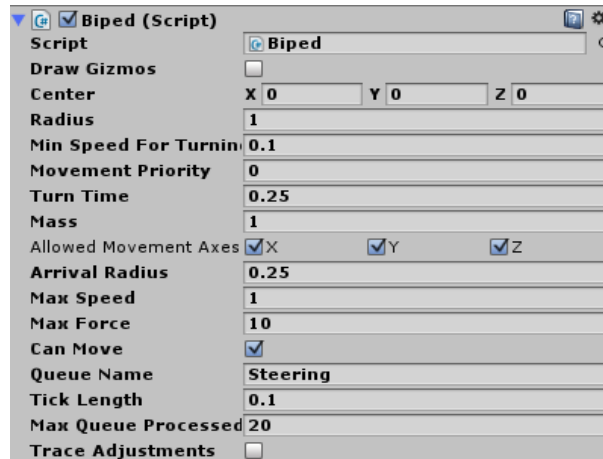


Figura 13 Controlador dos Soldados. Fonte: Próprio autor.

O grupo de variáveis responsável pelo controle dos comportamentos mantém-se inalterado logo que o processamento e resposta dos comportamentos forem iguais para ambos os modelos.

6.2 Steering Behaviors Individuais

Conforme apresentado na seção 4, os comportamentos individuais, Arrive e Obstacle Avoidance foram selecionados para serem implementados no projeto. Nesta seção a implementação destes comportamentos será descrita.

6.2.1 Arrive

Como visto na Figura 14, o algoritmo do Arrive apresenta as variáveis de peso (Weight), que é utilizada para dar pesos aos comportamentos, fazendo assim o movimento do agente ser mais ou menos influenciado pelo comportamento.

O objetivo (Target Point), que contém as coordenadas normais do alvo, é representado como um ponto no espaço tridimensional.

Os booleanos para considerar a velocidade (Consider Velocity), que determina o uso ou não da velocidade atual no cálculo do comportamento, e retorno a posição atual (Default To Current Position), que rotaciona o agente para a posição inicial assim que ele alcança o objetivo.

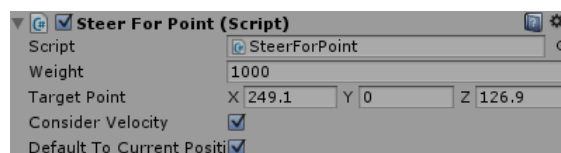


Figura 14 Script de Arrive. Fonte: Próprio autor.

Podemos observar na Figura 15 a força resultante do comportamento Arrive (linha azul) que direciona a LMU diretamente para o alvo (representado pelo quadrado branco). Nesse exemplo é considerada a velocidade da LMU, o que fez com que a mesma reduza a velocidade conforme se aproxima do alvo, tornando a parada mais verossímil a um veículo real.

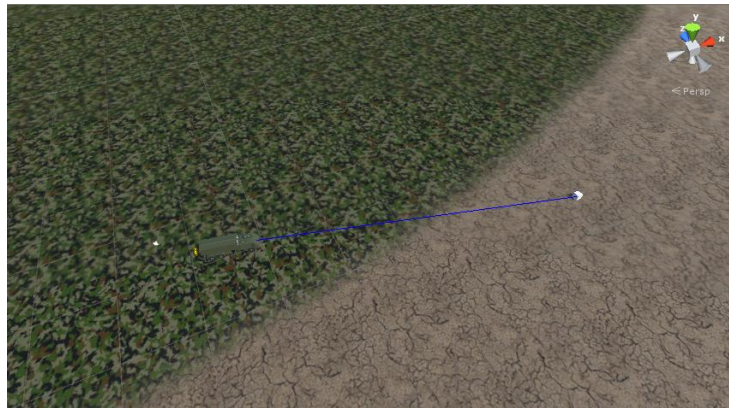


Figura 15 Arrive aplicado na LMU. Fonte: Próprio autor.

6.2.2 Obstacle Avoidance

O algoritmo de Obstacle Avoidance utiliza o mesmo detector que os algoritmos em grupo, que será discutido mais detalhadamente na seção seguinte, para avaliar os objetos que serão um obstáculo em seu trajeto até o objetivo.

Na Figura 16 pode ser observado que o algoritmo apresenta um peso (Weight) para calcular a importância do comportamento e uma variável para definir o tempo de estimativa (Estimation Time) que o comportamento usa para a previsão de movimento do agente e para detectar se ocorrerá uma colisão.

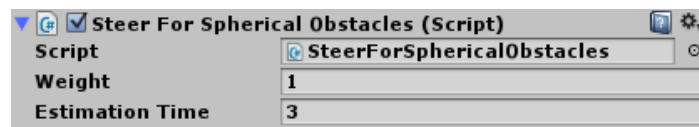


Figura 16 Script do Obstacle Avoidance. Fonte: Próprio autor.

O comportamento utiliza a visão do agente, representada na Figura 17 pela esfera ciano, para detectar algum obstáculo ou outro agente no caminho, representado pela esfera, que possui um raio de detecção (esfera azul escura em torno da esfera branca).

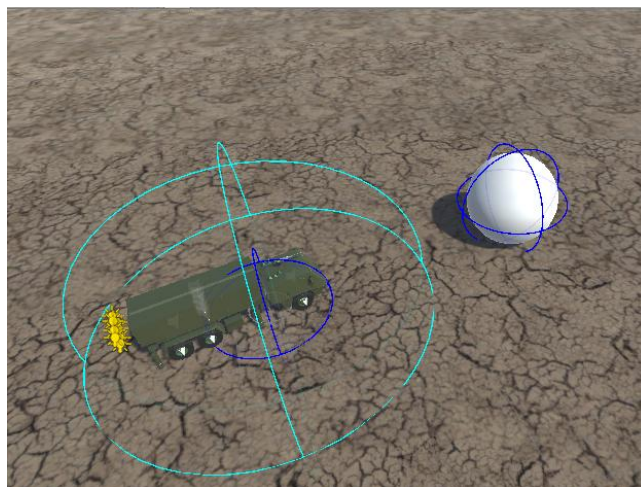


Figura 17 Obstacle Avoidance antes da detecção. Fonte: Próprio autor.

Na Figura 18 podemos observar que o agente detectou a esfera como um obstáculo em seu caminho. A esfera do obstáculo ficou vermelha, o que indica o

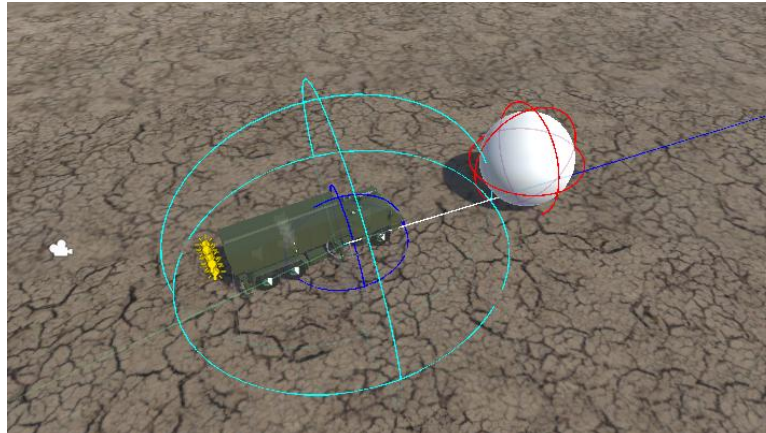


Figura 18 Detecção de obstáculo usando Obstacle Avoidance. Fonte: Próprio autor.

agente irá se colidir com a esfera caso siga trajeto atual, que é apresentado pela linha branca, antes da colisão, e azul, depois da colisão.

Na Figura 19 vemos o resultado do Obstacle Avoidance, onde depois de detectar a colisão é aplicada uma força de steering perpendicular ao caminho do agente e com intensidade proporcional ao tamanho do obstáculo, fazendo que ele rotacione e se desloque para o lado, evitando assim a colisão com a esfera. Então, como não mais ocorrerá uma colisão, sua esfera volta para a cor azul.

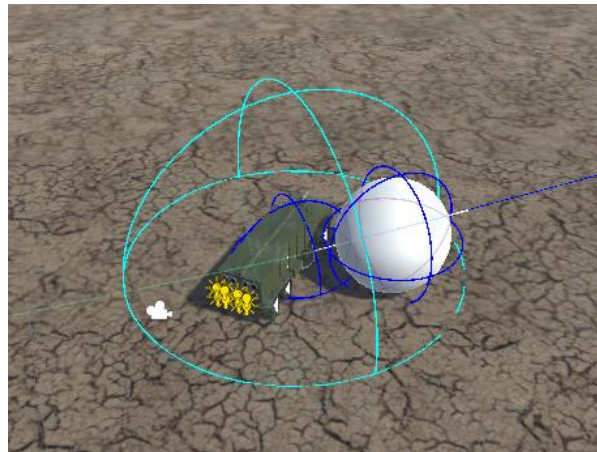


Figura 19 Agente efetuando o desvio calculado pelo Obstacle Avoidance. Fonte: Próprio autor.

6.3 Steering Behaviors de Grupo

Nessa seção serão explicadas as implementações dos algoritmos de grupo Cohesion, Alignment e Separation, bem como o método de detecção de vizinhos e obstáculos.

6.3.1 Reconhecimento de Vizinhos

Como apresentado na seção 2.1.2, esse algoritmo é o responsável pela análise da vizinhança do agente, detectando outros agentes ou obstáculos. Como visto na Figura 20, o script é composto pelas variáveis de controle de comportamentos, Queue Name, Max Queue Processed e Tick Length, que apresentam as mesmas funcionalidades dos scripts individuais, porém a Queue é renomeada para Radar, permitindo que sejam combinados os algoritmos de grupo e individuais.

O raio de detecção (Detection Radius) é a distância que o agente pode visualizar ao seu redor, tendo conhecimento dos obstáculos e agentes contidos nesse raio, demonstrado na Figura 21 pela esfera em ciano. Uma variável que assume valores verdadeiro ou falso, para detectar outros veículos (Detect Disabled Vehicles) que não estão ativos, evitando colisão entre dois agentes. Um seletor de camadas (Layers Checked), para selecionar quais camadas (UI, Terrain, Objects) serão detectadas pelo comportamento. E um buffer de objetos detectados por este agente (Pre Allocate Size), que armazena a referência de todos obstáculos e agentes dentro do raio de detecção.

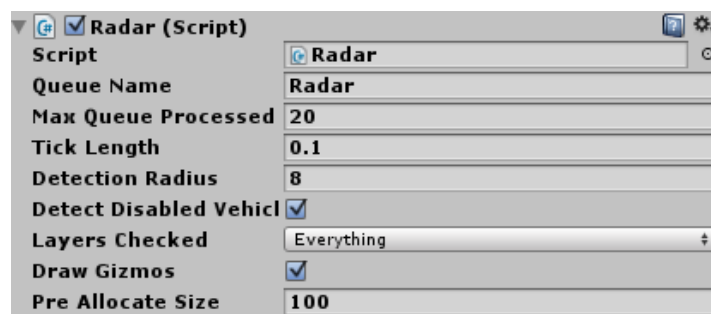


Figura 20 Script da detecção de vizinhos. Fonte: Próprio autor.

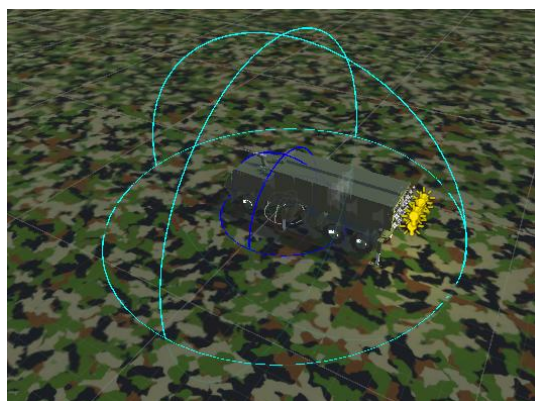


Figura 21 Raio de detecção do agente. Fonte: Próprio autor.

6.3.2 Cohesion

O comportamento Cohesion, como apresentado na Figura 22, também possui o peso (Weight) para sua consideração no resultado final do movimento. Além disso, apresenta a distância mínima (Min Distance) e distância máxima (Max Distance)

onde outros agentes devem estar para serem considerados um grupo e consequentemente relevantes no cálculo da força de atração resultante.

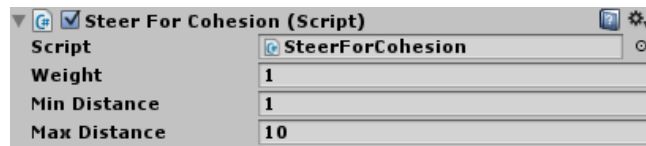


Figura 22 Script do Cohesion. Fonte: Próprio autor.

6.3.3 Separation

Já o comportamento Separation, como mostrado na Figura 23, possui mais variáveis que os outros comportamentos de grupo. Além da lista de agentes proveniente do radar e das variáveis de peso (Weight), distância mínima (Min Distance) e distância máxima (Max Distance), que apresentam as mesmas funcionalidades que o Cohesion, ele possui uma distância de conforto (Comfort Distance), que é a distância que um agente deve manter de outro agente. Um multiplicador dentro da zona de conforto (Multiplier Inside Comfort Zone), que irá servir como escalar de intensidade para a força repulsiva quando um agente entra na zona de conforto de outro. E o raio de impacto do agente (Vehicle Radius Impact), que determina a distância mínima obrigatória entre dois agentes para que evite a colisão entre eles.

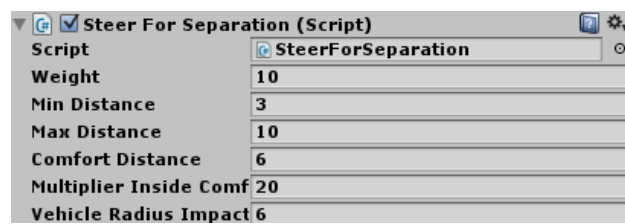


Figura 23 Script de Separation. Fonte: Próprio autor.

6.3.4 Alignment

Assim como o Cohesion, este comportamento usa a lista de agentes geradas pelo radar. Como mostra a Figura 24, este comportamento apresenta as variáveis de peso (Weight) na consideração da força resultante, a distância mínima (Min Distance) e distância máxima (Max Distance), utilizando a rotação dos agentes no grupo para retornar uma média de seus valores e então aplica a força para alinhar o agente aos demais do grupo.

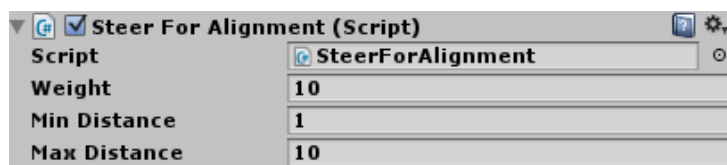


Figura 24 Script de Alignment. Fonte: Próprio autor.

Na Figura 25 podemos observar uma combinação dos comportamentos de grupo. Os 4 agentes (marruás) estão em um grupo, onde podemos perceber as esferas ciano representando a distância máxima de grupo, utilizada em todos

algoritmos, e a esfera azul representando a distância de conforto do algoritmo de Separation. Assim como todas elas estão direcionadas na mesma direção, devido ao uso do Alignment.

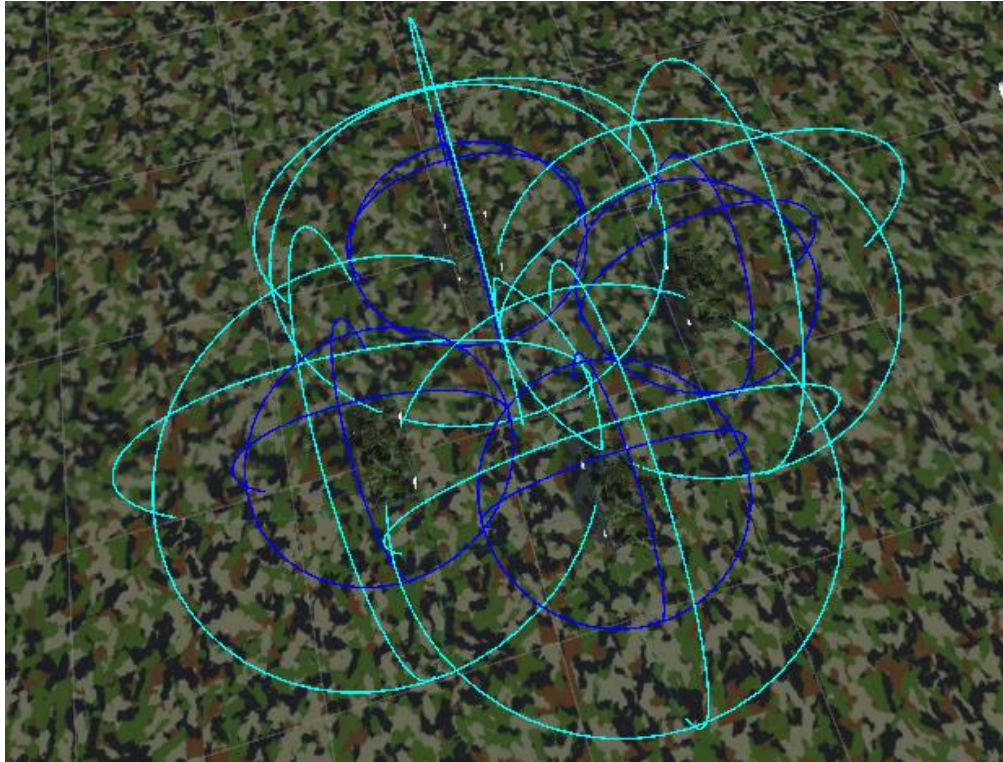


Figura 25 Combinação dos algoritmos de grupo. Fonte: Próprio autor.

6.4 Desafios Encontrados

Ao longo do desenvolvimento deste trabalho foram encontrados alguns desafios.

Primeiramente foi o uso dos comportamentos nos controladores utilizados no projeto. Devido ao fato dos controladores não apresentarem as variáveis necessárias para os algoritmos e não suportarem o uso de forças para seu controle, foi necessário então o desenvolvimento de outros controladores, que foram apresentados na seção 5.1, que solucionaram o problema.

O segundo desafio encontrado foi a normalização das forças resultantes. O movimento do agente ficava muito brusco, fazendo curvas com ângulos acentuados e percorrendo grandes distâncias em frações de segundos. Isso foi resolvido com a normalização das forças em relação a massa do agente.

Outro desafio encontrado foi a aplicação do Obstacle Avoidance em obstáculos não esféricos. Devido ao fato da fórmula utilizada apenas levar em conta a maior distância do centro do objeto a uma extremidade para o cálculo, ocorriam erros onde o agente desviava de objetos inexistentes. Foi desenvolvida uma adaptação para um objeto elíptico, porém não encontrou-se os resultados desejados, visto que ainda ocorriam casos de colisões e casos onde o agente desviava quando

não era necessário. Esse erro ainda não foi corrigido, devido ao fato da imprecisão de cálculo de colisões com objetos com formatos não elípticos.

Por último, a força resultante do Obstacle Avoidance acaba gerando uma curva não suavizada, o que torna o movimento pouco realista. Uma tentativa de contornar esse problema foi utilizando uma força de menor intensidade, essa abordagem não resolveu o problema e ocasionou colisões com obstáculos maiores que antes não ocorriam. Outra hipótese foi aplicar a força resultante apenas no eixo do veículo, mas não foi possível adaptar o controlador para que isso ocorresse. Este problema ainda não foi solucionado.

7 Resultados

Este capítulo aborda os resultados obtidos com o desenvolvimento dos comportamentos nos agentes utilizados no simulador do projeto SiS ASTROS 2020.

Na Figura 26 podemos comparar visualmente a precisão do Arrive, com a abordagem utilizada antes do desenvolvimento do steering behavior. A LMU com as esferas azul e ciano está com os comportamentos ativos, enquanto a outra possui apenas o controlador e utiliza o script provido pela Unity. Podemos perceber que a LMU com os comportamentos ficou em cima do alvo, enquanto a outra distanciou-se bastante do alvo.

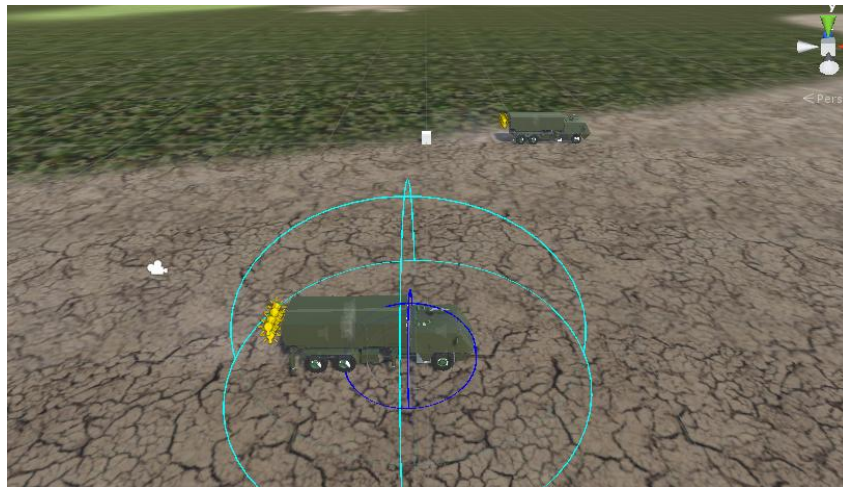


Figura 26 Comparação do Arrive com script da Unity3D. Fonte: Próprio autor.

Uma comparação entre a LMU com e sem Obstacle Avoidance é apresentada na Figura 27, onde a LMU sem os comportamentos implementados (modelo na posição superior da Figura) não consegue prosseguir devido a colisão com a esfera, enquanto a LMU com os comportamentos implementados (modelo na posição inferior da Figura) desvia do obstáculo e chega ao seu alvo.



Figura 27 Comparação do uso de Obstacle Avoidance Fonte: Próprio autor.

Uma comparação feita entre os comportamentos de grupo mostrou que sem o uso dos comportamentos desenvolvidos (Figura 28.1) as 3 LMUs acabam colidindo (Figura 28.2 e ao fim acabam ficando distantes do alvo de maneira desorganizada (Figura 28.3). Ao ativar os comportamentos de grupo nestes objetos observa-se que desde o início do trajeto as LMUs reconhecem o grupo e se mantem

alinhadas(Figura 29.1). Na Figura 29.2 podemos ver como elas se aproximam, mas ainda assim mantém a distância de conforto evitando uma colisão entre elas. Ao fim, na Figura 29.3, pode-se observar que o grupo de LMUs chegou ao alvo com precisão, onde as LMUs chegam ao objetivo sem colisões e alinhadas.



Figura 28 Grupo de LMUs sem comportamentos. Fonte: Próprio autor.

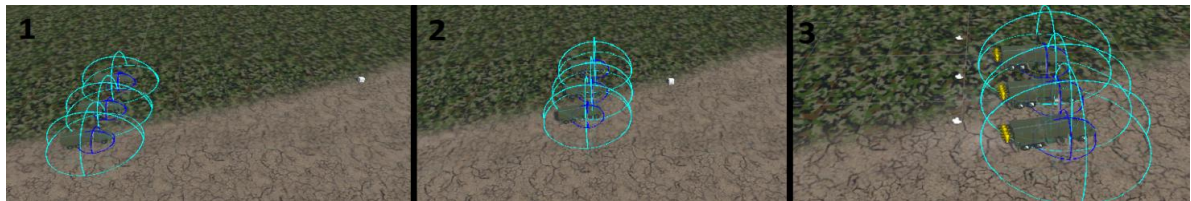


Figura 29 Grupo de LMUs com comportamentos. Fonte: Próprio autor.

8 Conclusão e Trabalhos Futuros

O presente trabalho teve como objetivo o desenvolvimento de Steering Behaviors para o projeto SiS ASTROS 2020 de forma a incrementar a qualidade visual e do funcionamento do simulador.

Analisando os resultados apresentados na seção anterior, podemos concluir que os objetivos desse trabalho foram alcançados, uma vez que as demonstrações foram satisfatórias.

As principais contribuições deste trabalho para o projeto são:

- Robustez dos agentes, evitando falhas por colisões com obstáculos do cenário ou com outros agentes.
- Agentes apresentam uma maior precisão em termos de aproximação de alvos.
- Melhora na capacidade de movimento dos agentes nos exercícios.

Por fim, são sugeridos como trabalhos futuros:

- A implementação do Obstacle Avoidance para obstáculos de diversas formas.
- Um estudo mais aprofundado para a suavização do desvio do Obstacle Avoidance, permitindo assim uma melhora visual no simulador.
- E a implementação desses algoritmos na versão oficial desse simulador.

9 Referências

Reynolds, C. W. Steering Behaviors For Autonomous Characters, in the proceedings of Game Developers Conference 1999 held in San Jose, California. Miller Freeman Game Group, San Francisco, California, 1999.

Reynolds, C. W. Interaction with Groups of Autonomous Characters, in the proceedings of Game Developers Conference 2000, CMP Game Media Group (formerly: Miller Freeman Game Group), San Francisco, California, 2000.

Balch, Tucker, Arkin, R.C, Behavior-based Formation Control for Multi-robot Teams IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 1999.

Go Jared. Vu ,Thuc D. Kuffner, James J. Autonomous *behaviors* for interactive vehicle animations. Science Direct, Pittsburgh , 2005.

Fajen , Brett R. Warren, William H. *Behavioral Dynamics of Steering, Obstacle Avoidance, and Route Selection.* 2003.

White, K. Preston Jr. Ingalls, Ricki G. INTRODUCTION TO SIMULATION. Proceedings of the 2009 Winter Simulation Conference, Virginia 2009.

Hartley III ,Dean S. VERIFICATION & VALIDATION IN MILITARY SIMULATIONS. Proceedings of the 1997 Winter Simulation Conference. Virginia, 1997.

Garcia, Maj Cav Flávio dos Santos Lajóia. O emprego da simulação de combate como ferramenta de apoio ao projeto organizacional e doutrinário da Força Terrestre Brasileira. ECEME, Rio de Janeiro, 2006.

Pereira Junior, Sergio Manoel Martins. A SIMULAÇÃO DE COMBATE APLICADA AO ENSINO, À INSTRUÇÃO E AO ADESTRAMENTO DAS TROPAS BLINDADAS DO EB. ECEME, 2008.

CUNHA, MAJ ENG ANDRÉ LUIZ NOBRE. O Emprego do Sistema de Simulação Construtiva como Ferramenta de Apoio à Decisão: uma proposta ao Exército Brasileiro. ECEME ,Rio de Janeiro, 2011.

Crookal, David. Towards a reconceptualization of simulation: from representation to reality. Oxford, 1987.

Wang, Hongling. Kearney, Joseph K. Cremer, James. Willemsen, Peter. Steering Behaviors for Autonomous Vehicles in Virtual Environments. IEEE Virtual Reality, Alemanha. 2005.

Maes, Pattie. Artificial Life meets Entertainment: Lifelike Autonomous Agents. MIT Press, Cambridge, Mass. 1995.

Unity, <http://unity3d.com/pt/unity>.

UnitySteer, disponível em <https://github.com/ricardojmendez/UnitySteer>