

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA - CT
CURSO DE CIÊNCIA DA COMPUTAÇÃO - CCC**

**DESENVOLVIMENTO DE UM
CONTROLADOR AUTÔNOMO PARA
VEÍCULOS EM UM AMBIENTE DE
SIMULAÇÃO**

TRABALHO DE GRADUAÇÃO

Vítor Peixoto Menezes

Santa Maria, RS, Brasil

2016

DESENVOLVIMENTO DE UM CONTROLADOR AUTÔNOMO PARA VEÍCULOS EM UM AMBIENTE DE SIMULAÇÃO

Vítor Peixoto Menezes

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

Orientador: Prof. Dr. Cesar Tadeu Pozzer

**Trabalho de Graduação N° 430
Santa Maria, RS, Brasil**

2016

Menezes, Vítor Peixoto

Desenvolvimento de um controlador autônomo para veículos em um ambiente de simulação / por Vítor Peixoto Menezes. – 2016.

41 f.: il.; 30 cm.

Orientador: Cesar Tadeu Pozzer

Monografia (Graduação) - Universidade Federal de Santa Maria, Centro de Tecnologia - CT, Curso de Ciência da Computação - CCC, RS, 2016.

1. Controlador autônomo de veículos. 2. TCC. 3. Steering Behaviour para veículos. I. Pozzer, Cesar Tadeu. II. Título.

© 2016

Todos os direitos autorais reservados a Vítor Peixoto Menezes. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: vmenezes@inf.ufsm.br

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


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

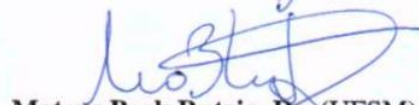
**DESENVOLVIMENTO DE UM CONTROLADOR AUTÔNOMO PARA
VEÍCULOS EM UM AMBIENTE DE SIMULAÇÃO**

elaborado por
Vítor Peixoto Menezes

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

COMISSÃO EXAMINADORA:


Cesar Tadeu Pozzer, Dr.
(Presidente/Orientador)


Mateus Beck Rutzig, Dr. (UFSM)


Alex Thomas Almeida Frasson, Bel. (UFSM)

Santa Maria, 16 de Dezembro de 2016.

Dedico este trabalho aos meus pais Cláudio Carvalho Menezes e Maria Cristina de C. P. Menezes e a meu irmão Rafael Peixoto Menezes pois serviram de base para a minha vida e deram todo suporte para a minha formação.

AGRADECIMENTOS

Agradeço especialmente meus pais Cláudio Carvalho Menezes e Maria Cristina de C. P. Menezes, pelo amor, incentivo e apoio incondicional. Aos exemplos que foram durante toda a minha vida, mostrando o que é certo e errado e a lutar pelos meus objetivos. Meu irmão Rafael Peixoto Menezes, companheiro para toda hora, que estava lá em todos os momentos igualmente apoiando e torcendo.

Meus agradecimentos à todos os amigos que foram de vital suporte, companheiros de trabalho e irmãos na amizade que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza. Alguns se mudaram, outros continuam presentes, mas todos de igual importância. Gostaria de agradecer especialmente à Isabela Thones Ceretta por ser um importante apoio para a conclusão do trabalho. Um agradecimento especial ao Prof. Dr. Cesar Tadeu pozzier pela paciência na orientação e incentivo que tornaram possível a conclusão deste trabalho.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação - CCC
Universidade Federal de Santa Maria

DESENVOLVIMENTO DE UM CONTROLADOR AUTÔNOMO PARA VEÍCULOS EM UM AMBIENTE DE SIMULAÇÃO

AUTOR: VÍTOR PEIXOTO MENEZES

ORIENTADOR: CESAR TADEU POZZER

Local da Defesa e Data: Santa Maria, 16 de Dezembro de 2016.

Um controlador autônomo para veículos em um ambiente de simulação virtual visa suprir as necessidades de movimentação e comportamento dos veículos nos trajetos escolhidos dentro dos ambientes fornecidos, gerenciando as ações que o automóvel irá realizar em situações na qual se dispõe no cenário. Este trabalho apresenta o desenvolvimento de um controlador para governar a movimentação e o comportamento padrão dos veículos inseridos no meio, para que possam responder às ordens dadas aos mesmos. Para a solução do problema proposto foram utilizadas técnicas de navegação de agentes autônomos em ambientes simulados, modelos de movimento em grupo e soluções para o desvio de obstáculo em tempo real.

Palavras-chave: Controlador autônomo de veículos. TCC. Steering Behaviour para veículos.

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Descrição do Problema	10
1.2 Objetivos	10
2 REVISÃO TEÓRICA	12
2.1 Steering Behaviour	12
2.1.1 Desvio de Obstáculo	12
2.1.1.1 Verificando o Caminho	14
2.1.1.2 Verificando Colisão	14
2.1.1.3 Calculando as Forças de Desvio.....	15
3 REVISÃO BIBLIOGRÁFICA	17
3.1 Planejamento de Caminhos em Tempo Real	17
3.2 Fluxo de Informações e Controle Cooperativo de Formação de Veículos	19
3.3 Líderes Virtuais, Potenciais Artificiais e Controle de Grupo Coordenado	21
4 ARQUITETURA	23
5 IMPLEMENTAÇÃO	24
5.1 Unity3D	24
5.1.1 Rigidbody	24
5.1.2 Colliders	25
5.1.2.1 Wheel Collider	25
5.1.3 Centro de Massa	25
5.1.4 Módulo da física	27
5.1.4.1 Forças - Aceleração e Frenagem	27
5.1.4.2 Rotação das rodas e Barra estabilizadora	27
5.1.4.3 Representação Visual das Forças Aplicadas nas Rodas e Eixos	29
5.2 Módulo do comportamento	29
5.2.1 Controle de velocidade atual	30
5.2.2 Seguir Fielmente o Caminho	31
5.2.3 Deslocamento em Comboio	32
5.2.4 Garantir a ordem das viaturas dentro do comboio	33
5.2.5 Comportamento de Redirecionamento	33
5.2.5.1 Sensores	33
5.2.5.2 Variação das Forças do Steering Behaviour para Utilização em Veículos	34
6 RESULTADOS	36
7 CONCLUSÃO	39
REFERÊNCIAS	40

1 INTRODUÇÃO

Reproduzir a movimentação de agentes controlados por inteligência artificial é de extrema importância para a fidelidade de ambientes virtuais. Quando há objetos controlados por um comportamento autônomo em um ambiente virtual, os mesmos criam a sensação de um realismo dinâmico. O principal objetivo do controlador é gerenciar de maneira inteligente o deslocamento de uma ou mais viaturas distribuídas organizadamente em um cenário simulado. Tal procedimento é complexo devido à diversas situações que devem ser tratadas.

O controlador autônomo de veículos é uma necessidade do projeto Sistema Integrado de Simulação Astros (SIS-ASTROS) e visa simular a movimentação de viaturas em um cenário virtual 3D. O sistema é constituído de diversas viaturas responsáveis pelo reconhecimento de terreno, carregamento de munições, controle meteorológico e lançamento de mísseis. O sistema servirá não só para simular ambientes, mas também para definir de que forma devem ser organizadas as viaturas enquanto estiverem sendo utilizadas.

A simulação visa treinar alunos em doutrinas de Reconhecimento, Escolha e Ocupação de Posição (REOP). Assim, dentro da simulação é construída uma situação em que aluno utilizará uma mesa tática digital para atuar como Capitão de Bateria e fornecer ordens para a bateria. Os veículos utilizarão o controlador para executar às ordens de movimentação dadas pelo aluno como estacionar, andar de forma independente ou andar em comboio.

Para um veículo automotor navegar entre dois pontos, é necessário obter diversas informações referentes à sua redondeza, uma vez que a rota definida pode estar bloqueada por um obstáculo. A solução para esse caso é construir um algoritmo robusto de controle de manobras que trate a situação de desviar do mesmo. Outra funcionalidade necessária no controlador é cursar fielmente um dado itinerário, pois é de extrema importância para a reprodução do comportamento correto da doutrina utilizada pelo exército que o veículo se mova exatamente sobre a rota traçada.

Um veículo autônomo deve combinar aspectos de um robô, com a habilidade de improvisar em situações adversas, escolhendo a opção adequada para sua visão do ambiente. O mesmo pode existir isolado ou em uma situação compartilhada por outras entidades. Assim, há a necessidade de "enxergar" o arredor para entender a situação atual. Tendo-se um caminho definido por um algoritmo de path-finding, o controlador autônomo deve chegar ao destino gerenciando todos os aspectos de veículos, como velocidade, curvas, direção, obstáculos, entre

outros.

A representação do ambiente tem um impacto muito grande para o planejamento do comportamento autônomo. Assim, para o contexto do projeto SIS-ASTROS utilizou-se o motor gráfico Unity3D, baseando-se nas bibliotecas fornecidas para reprodução física. O algoritmo de path-finding A* foi introduzido para fazer a geração do melhor caminho dado um ponto inicial e um ponto final, dentro de um ambiente estático. Gerenciar o veículo de forma autônoma recebendo de entrada somente um percurso e chegar ao seu final de forma eficiente e é em suma o objetivo geral do controlador.

1.1 Descrição do Problema

A reprodução da inteligência humana no controle de um automóvel é uma tarefa complexa e é normalmente necessária ao se trabalhar com simulação onde há a presença de veículos controlados artificialmente. A eficácia da simulação depende diretamente do detalhamento dos eventos e do comportamento de seus agentes autônomos, assim, a importância de os autômatos terem um comportamento apropriado em todas as situações é imprescindível.

Dependendo da simulação, as situações que podem ocorrer são previsíveis e podem ser contornadas com o tratamento específico de cada uma. A necessidade dos veículos no projeto foram seguir corretamente um dado itinerário, controlando a velocidade máxima, mínima, distância, direção, andar em comboio, andar na ordem correta dentro do comboio e não colidir com outros veículos. Esses tópicos são complexos, uma vez que são implementados utilizando bases físicas, aplicando forças para o movimento das viaturas.

1.2 Objetivos

Este trabalho visa suprir as necessidades da automatização dos veículos no simulador SiS-ASTROS fazendo uso da engine gráfica Unity. A API que a Unity oferece auxilia a coleta das informações referentes ao terreno e objetos estáticos para realizar a navegação para seu destino, que são tarefas imprescindíveis para a confecção do controlador. Há algumas necessidades visuais e físicas, como reproduzir fielmente a colisão, suspensão, equilíbrio do veículo, velocidade máxima, tração, entre outros, porém, o objetivo principal é com seu comportamento.

Os objetivos específicos são:

- Reproduzir a física do veículo.

- Incluir funcionalidades básicas do motor.
- Desenvolver uma solução simples e elegante para os problemas mais comuns de navegação.
- Implementar o controlador de forma que possa ser reutilizado em qualquer tipo de veículo.

2 REVISÃO TEÓRICA

No presente capítulo será apresentado o conceito que auxiliou a confecção do controlador autônomo. A implementação do controlador é fundamentada principalmente na técnica de steering behaviour. Essa abordagem visa gerenciar o veículo utilizando uma visão local na tentativa de encontrar uma direção na qual o automóvel possa cruzar.

2.1 Steering Behaviour

Essa técnica visa permitir objetos autônomos se moverem de forma realista usando um esquema simples de forças, que combinadas produzem o resultado de desviar de objetos. A principal característica dessa abordagem é a utilização de simples forças locais baseadas nos vizinhos, em vez de recalculando caminho em tempo-real (SUD et al., 2008), por exemplo. Essa abordagem, mesmo aparentemente simples, é capaz de produzir padrões de movimentos complexos utilizando pouco poder de processamento.

A implementação original (REYNOLDS, 1999) e sua primeira formulação (REYNOLDS, 1987) mostra que pode-se simular quaisquer comportamentos simplesmente somando as forças resultantes num vetor resposta e somá-lo com a velocidade instantânea do agente. Assim, como as necessidades do simulador SIS-ASTROS requisitou somente o comportamento desvio de obstáculo, uma vez que os outros aspectos como perseguir e seguir caminho, têm especificidades que não condizem com a implementação original e foi necessário modificá-los completamente ou parcialmente. Os tópicos seguintes apresentaram algumas de suas características relevantes. A figura 2.1 mostra de maneira genérica o funcionamento da técnica.

2.1.1 Desvio de Obstáculo

A ideia básica por trás do desvio de obstáculos é gerar uma força suficiente para desviar do objeto cada vez que algum esteja perto o suficiente para bloquear o caminho. Mesmo que o ambiente tenha diversos bloqueios, esse comportamento vai utilizar uma obstrução por vez para calcular a força necessária para o desvio, assim, não havendo nenhum tipo de planejamento.

Somente as obstruções que estão na frente do personagem são analisados, o mais perto é dito como o mais perigoso, e é selecionado para avaliação. Como resultado o agente é capaz de desviar de todos os impedimentos na área, fazendo transições de um obstáculo para o outro

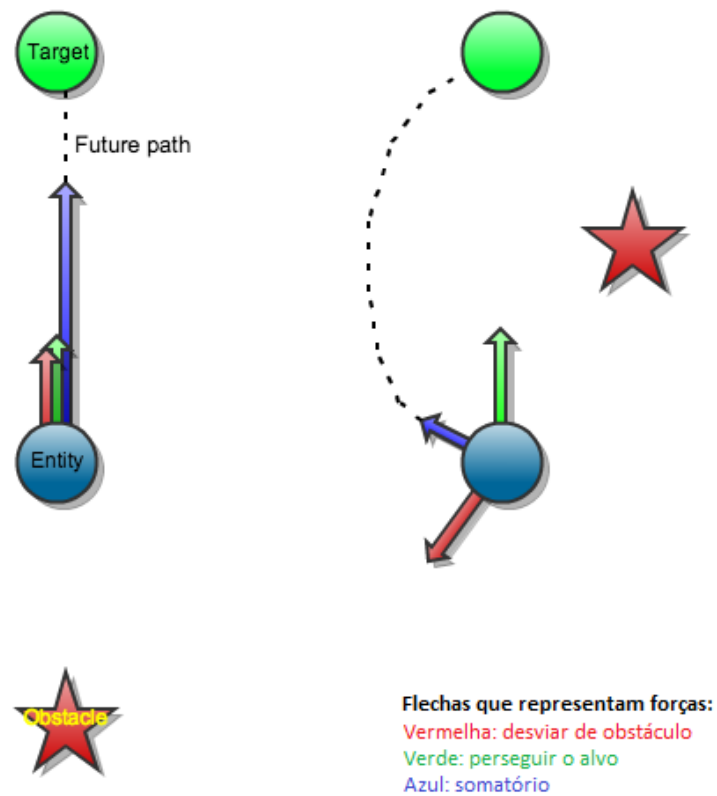


Figura 2.1 – Steering Behaviour

escolhendo sempre o mais próximo.

O desvio de obstáculo não é um algoritmo de path-finding. Ele somente irá tomar o controle do agente e fazê-lo modificar sua direção para não colidir até ultrapassar os bloqueios e retornar ao caminho original. Há alguns casos onde o desvio não gera um resultado ótimo, que são obstruções em formato de "L" ou "T". A figura 2.2 exemplifica o desvio de obstáculos numa cena, levando em conta a obstrução mais perigosa.

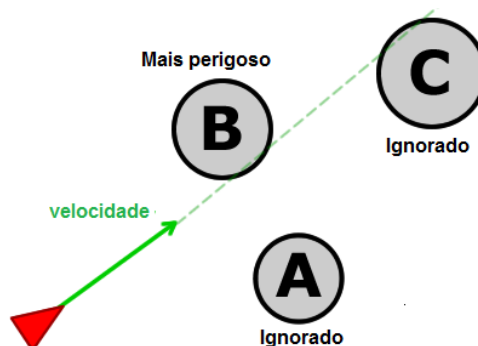


Figura 2.2 – Desvio de Obstáculo

2.1.1.1 Verificando o Caminho

O primeiro passo para desviar de um obstáculo é verificar a existência de objetos bloqueando o caminho. Os únicos objetos que o agente deve procurar são os que bloqueiam o caminho a sua frente (REYNOLDS, 1999). Para isso, verifica-se, em linha reta e na direção do movimento, bloqueios a uma dada distância.

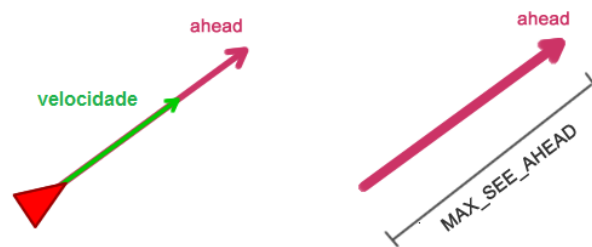


Figura 2.3 – Verificando o Caminho

O tamanho do vetor ahead (ajustado pela variável MAX_SEE_AHEAD) define o quão longe o agente vai enxergar. Quanto maior MAX_SEE_AHEAD, mais cedo o agente vai começar a desviar do obstáculo pois irá percebê-lo como uma ameaça antes como destaca a figura 2.3.

O vetor é calculado da seguinte forma:

```
ahead = myPosition + myForward * MAX_SEE_AHEAD;
```

2.1.1.2 Verificando Colisão

Para se verificar qualquer colisão, é necessário descrever os objetos em uma forma geométrica e representá-los com um colisor. Para se obter melhores resultados no desvio e evitar obstruções onde o agente não encontre o caminho, é aproximado a geometria para a menor esfera (círculo em duas dimensões) que engloba o objeto, então todos os obstáculos serão representados por uma esfera, assim, evitando bloqueios em formato de "L" ou "T".

Uma possível solução é detectar por uma colisão linha-circunferência, utilizando como linha o vetor ahead e a menor esfera que engloba o obstáculo como mostra a figura 2.4. Se a

distância for menor ou igual ao raio da esfera, então o vetor está dentro da esfera e uma colisão foi encontrada.

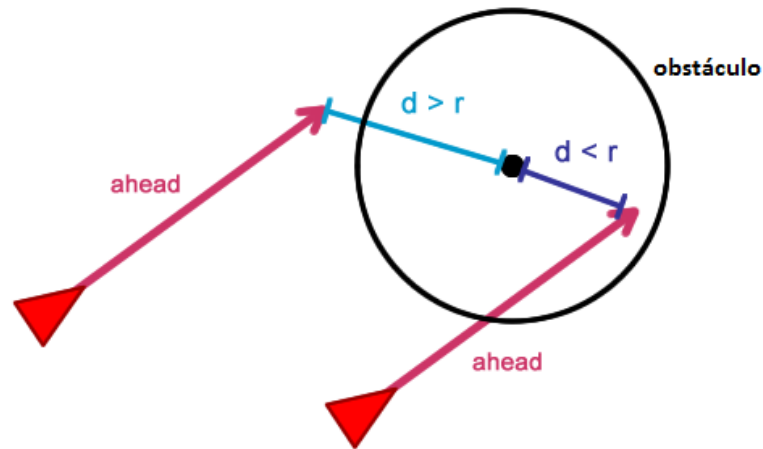


Figura 2.4 – Verificando colisão

2.1.1.3 Calculando as Forças de Desvio

A força deve empurrar o personagem para longe dos obstáculos, permitindo que o mesmo desvie da circunferência. Para encontrar o vetor de desvio, usa-se o centro da esfera (que é um vetor posição) e o vetor ahead. Podemos calculá-la da seguinte maneira:

```
avoidance_force = ahead - obstacle_center;
```

```
avoidance_force = avoidance_force.normalized * MAX_AVOID_FORCE;
```

Assim, `avoidance_force` é calculado, e logo em seguida normalizado e escalado por `MAX_AVOID_FORCE`. Quanto maior for `MAX_AVOID_FORCE`, mais forte será a força empurrando o agente para longe do obstáculo.

Para desviar do objeto, basta somar as forças calculadas no vetor velocidade final do agente como mostra a figura 2.5. Como explicado anteriormente, todas as forças são combinadas em uma, produzindo uma força que representa todo o comportamento. Essas forças são aplicadas de maneira instantânea no objeto fazendo com que o desvio seja imediato, de maneira brusca.

Essa abordagem supre a necessidade do simulador pois é uma estratégia local, de sim-

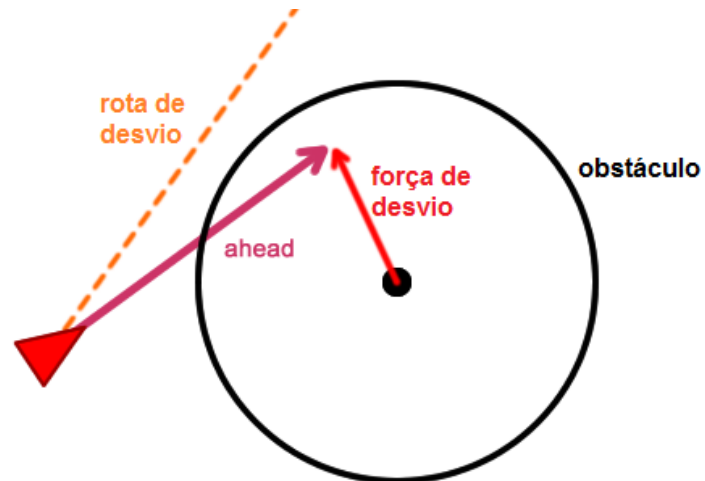


Figura 2.5 – Forças de Desvio

ples implementação e resultados eficientes para os obstáculos presentes no simulador. Pode-se aperfeiçoar a técnica para representar objetos e cenários mais complexos, de acordo com as situações apresentadas pelo simulador, de modo que não impacte negativamente em outras atividades realizadas.

3 REVISÃO BIBLIOGRÁFICA

No presente capítulo serão apresentadas alternativas relacionadas a controladores autônomos e técnicas que motivaram a escolha e resolução do problema proposto. Deve-se levar em conta que o simulador SIS-ASTROS deve dispôr de um veículo autônomo capaz de andar sob um percurso fielmente e conseguir desviar de obstáculos de maneira autônoma e local, ou seja, verificando somente sua vizinhança e tomando decisões à partir da mesma.

3.1 Planejamento de Caminhos em Tempo Real

Desviar de obstáculos de uma maneira eficiente e inteligente não é um trabalho realizado pelo desvio de obstáculos do Steering Behaviour. Como dito anteriormente, o Steering Behaviour não possui nenhum tipo de planejamento de caminho, podendo desviar para uma direção que não há saída para o objetivo, mesmo que haja um caminho mais eficiente desviando para a outra direção.

Segundo (SUD et al., 2008) uma alternativa que soluciona o problema de encontrar um caminho até um destino é computar para múltiplos agentes e tratar colisões planejando a rota para encontrar o objetivo ajustando individualmente cada agente. Como cada indivíduo ajusta seu comportamento de acordo com fatores dinâmicos do ambiente, técnicas que modelam o comportamento individual oferecem benefícios mais atrativos. Um dos principais desafios para simulação de agentes em grande escala é o planejamento global de grupo para cada agente.

O planejamento de caminho pode se tornar um grande obstáculo ao contabilizar um grande número de unidades virtuais se movimentando ao mesmo tempo de forma dinâmica. Muitas das técnicas já conhecidas utilizadas para calcular um caminho ou restringem para um ambiente estático ou fazem cálculos locais de colisão para reduzir a computação necessária. Esse último pode resultar em comportamento não natural ou não conseguir encontrar um caminho em uma obstrução de geometria complexa e vagar indefinidamente em um região. Esses problemas tendem a ser mais frequentes e visíveis em cenas dinâmicas com diversos agentes se movimentando.

A solução proposta visa calcular caminhos em tempo real para múltiplos agentes em um ambiente dinâmico. É introduzido um novo tipo de dado chamado Multi-Agent Navigation Graph(MaNNG) e utiliza GPU para computar diagramas discretos de Voronoi de primeira e se-

gunda ordem. O voronoi de primeira ordem calcula os pontos mais próximos a um agente. O voronoi de segunda ordem abrange um par de agentes e calcula os pontos mais próximos. Os diagramas de voronoi tem sido bastante utilizados para computação de planejamento de caminhos em ambientes estáticos (LATOMBE, 2012; CHOSSET, 2005) e estamos extendendo-o para ambientes dinâmicos.

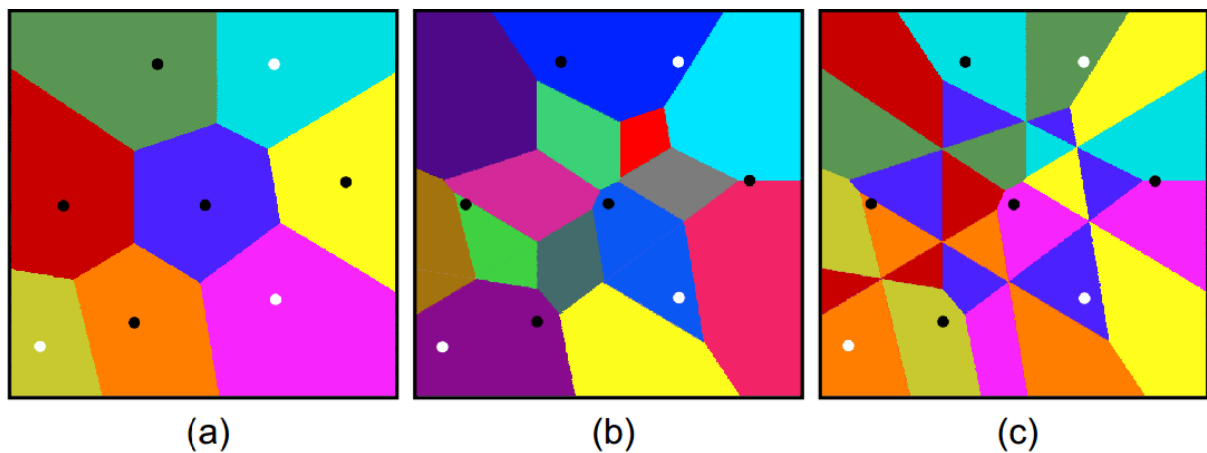


Figura 3.1 – (a) Diagrama de voronoi de 1ª ordem, (b) Diagrama de voronoi de 2ª ordem, 2º diagrama do vizinho mais próximo.

Os diagramas de voronoi, representados na figura 3.1, codificam as conectividades de espaço e fornecem um caminho com maior espaçamento para um agente. Para reproduzir uma cena dinâmica utilizando essa solução, é aproximado o voronoi para cada agente tratando outros agentes como obstáculos. Essa aproximação é muito custosa a medida que o número de agentes aumenta. Em vez disso, é computado somente os diagramas de voronoi de primeira e segunda ordem de todos os obstáculos e agentes e, assim, fica claro que há informações suficientes e consistentes para todos os agentes simultaneamente. Assim, é combinado diagramas de voronoi de primeira e segunda ordem para computar a MaNG para um planejamento global de agentes virtuais.

Como mostra a figura 3.2, o MaNG computa caminhos com maior espaçamento para um grupo de agentes móveis com diferentes destinos simultaneamente e não necessita uma estrutura de dado separada para planejar o caminho para cada agente. É computado uma aproximação discreta para essa estrutura usando a GPU e é proposto uma técnica de culling adaptativo para acelerar a computação. É também ajustado os problemas de subamostragem devido à discretização.

Essa abordagem trata movimentos complexos de várias viaturas se movimentando numa mesma região e planejamento coordenado. Porém, a solução não se encaixa no modelo atual e

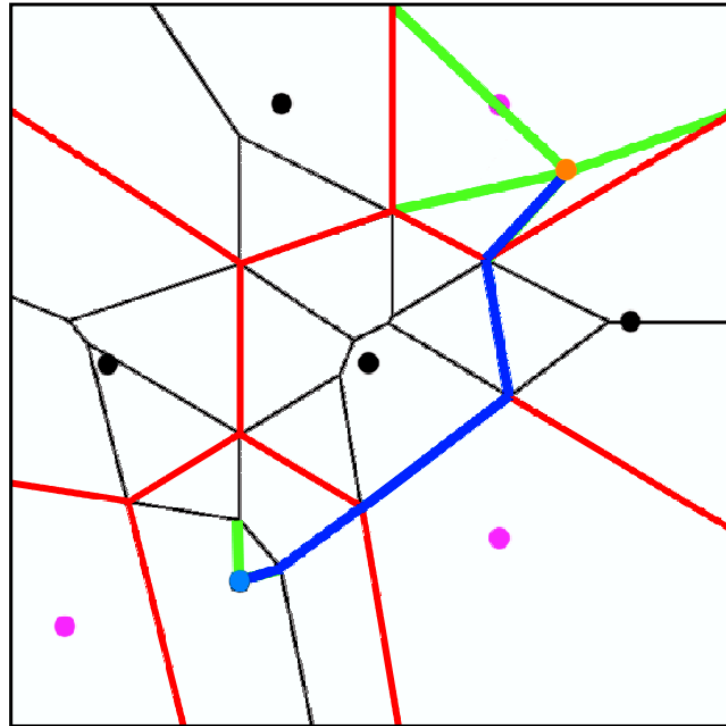


Figura 3.2 – Em linhas e pontos pretos está representado o diagrama de voronoi. O MaNG está representado por arestas verdes que conectam posições iniciais a posições finais. A posição inicial está representado por um ponto azul. A posição destino é representado por um ponto laranja. O menor caminho computado são as arestas em azul.

nas necessidades do simulador, uma vez que não temos prioridade do poder de processamento para dar ênfase para encontrar um caminho ótimo para as viaturas, mas é uma abordagem que pode ser implementada parcialmente, reduzindo os custos computacionais. Há outras soluções que não utilizam tanto poder computacional e oferecem respostas satisfatórias para o problema.

3.2 Fluxo de Informações e Controle Cooperativo de Formação de Veículos

Abordagens existentes no controle de formação de veículos são geralmente divididas em duas categorias. A primeira é a abordagem "líder-seguidor". Essa tem vantagens de simplicidade referente a trajetória clara definida pelo líder e que a estabilidade das formações é dirigida pela estabilidade individual das regras impostas nos veículos. Porém, essas soluções são conhecidas por tratamentos a imprevistos insatisfatórias (YANAKIEV; KANELLAKOPOULOS, 1996). Adicionalmente, essa estratégia depende fortemente do líder encontrar o objetivo, e, confiar exclusivamente em um indivíduo pode ser indesejável, especialmente em ambientes adversos. A segunda abordagem é "líderes virtuais" (SMITH; HANSSMANN; LEONARD, 2001; LEONARD; FIORELLI, 2001; EGERSTEDT; HU; STOTSKY, 2001), nos quais a formação

de veículo é clustorizada em um ou mais veículos fictícios nos quais as trajetórias agregadas atuam como líder. Essa abordagem resolve problemas de perturbação, falhas, mas necessita de comunicação intensa entre o grupo para estabelecer as posições e propriedades desses líderes.

A solução proposta por (FAX; MURRAY, 2004) visa corrigir o problema de cooperação de um grupo de veículos realizando uma tarefa compartilhada usando comunicação entre os mesmos para coordenar suas ações. A intenção é de considerar um número maior de possibilidades de conexões entre veículos e entender como a topologia do fluxo de informação afeta a estabilidade e o desempenho do sistema enquanto é realizado uma tarefa em conjunto. Em suma, desenvolver estratégias de troca de informação para aprimorar a estabilidade e o desempenho, sendo robustos para trocas na topologia de comunicação.

Essa abordagem proposta é um modelo de comunicação utilizando uma topologia parecida com a de um grafo, vide figura 3.3. Cada veículo é um nó direcionado e há uma conexão do nó i para o nó j caso o veículo i receba informação do veículo j . Juntando a teoria dos grafos, teoria de controle e teoria de sistemas dinâmicos, foi estudado a interação entre a rede de comunicação e a dinâmica do veículo para propor estratégias de troca de informação. Nesse contexto, é dado condições necessárias e suficientes para a estabilidade de um sistema interconectado de veículos idênticos.

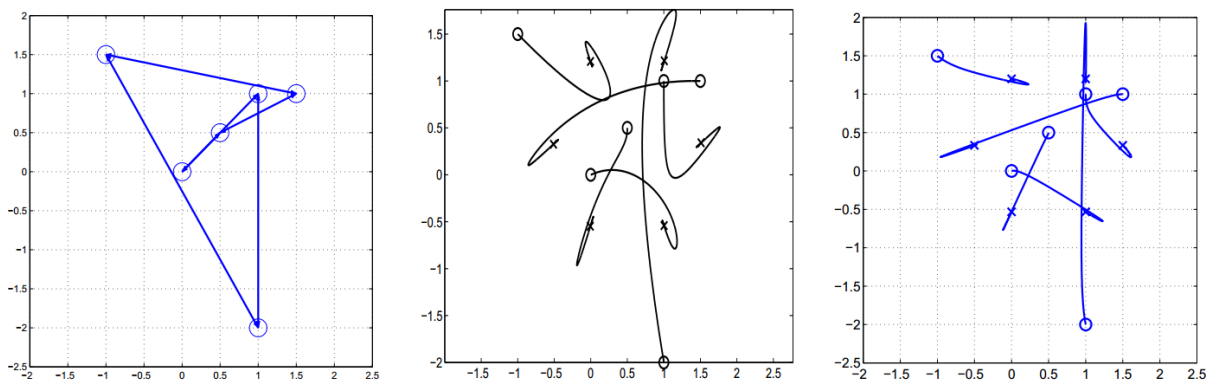


Figura 3.3 – (a) Condição inicial e verificação de topologia, (b) Informações capturadas, (c) Fluxo de informações verificadas e enviadas.

Essa abordagem abrange um fluxo de informação bastante robusto para a comunicação entre veículos. Essa abordagem pode ser necessária em casos de grupos diferentes de comboios em proximidades, assim, gerando cooperação entre os mesmos. Pode-se utilizar essa solução para controlar a movimentação de um grupo de veículos independentes, cenário no qual ocorre dentro do simulador.

3.3 Líderes Virtuais, Potenciais Artificiais e Controle de Grupo Coordenado

A abordagem de (LEONARD; FIORELLI, 2001) propõe uma aproximação de controle distribuído para coordenar múltiplos veículos autônomos baseados em potenciais artificiais e líderes virtuais. Um objetivo central é utilizado para contribuir para a metodologia para sintetizar robustas e escaláveis regras de controle relativamente simples no nível de indivíduo que permitam que veículos dentro de um grupo agirem com maior desempenho e inteligência.

A ideia visa simular o comportamento local no nível individual. Isso inclui as habilidades de manobras coordenadas de forma rápida e eficiente, de processar rapidamente dados e uma significativa melhora no habilidade de fazer decisão.

O uso de potenciais artificiais na aproximação é inspirada pelas observações e modelos de biólogos observando grupos de animais em bandos exibindo características coordenadas. Grupos na natureza usam um controle distribuído em resposta a seus sentidos individuais em relação ao ambiente e são regidos pelo comportamento dos vizinhos. Biólogos sugerem que os seguintes elementos são básicos para manter uma estrutura de grupo: (1) atração para vizinhos distantes dado uma distância máxima, (2) repulsão de vizinhos muito próximos e (3) alinhamento ou velocidade correspondente com os vizinhos.

A framework proposta por (LEONARD; FIORELLI, 2001), codifica essas regras locais de tráfego com as potenciais artificiais que definem a interação dessas forças entre veículos vizinhos. Cada um desses potenciais é uma função da distância relativa de pares de veículos. Nesse estágio as forças de controle para um indivíduo podem ser derivada pelo gradiente negativo da soma de todos os potenciais afetando aquele veículo. Os potenciais locais podem ser designados para corresponder a uma geometria de grupo desejada com valores predefinidos de espaço entre veículos. A função de Lyapunov para provar a estabilidade e a robustez que um movimento em grupo pode ser construída como a soma das energias cinéticas dos veículos e as energias potenciais artificiais.

Além dos campos potenciais entre os veículos, é introduzido campos potenciais locais associados com o ponto de referência do movimento que é chamado de líderes virtuais ou guias virtuais (que não são veículos). Os veículos reagem aos guias virtuais da mesma maneira que vizinhos reais. A proposta de líderes virtuais é introduzir direção, bando e/ou manipular o comportamento do grupo de veículos.

É enfatizado que não tem líderes entre os veículos. Na verdade, não há necessidade de ter

ordem entre eles; qualquer veículo pode ultrapassar outro. Essa característica da aproximação proposta adiciona robustez ao grupo contra uma falha de um único veículo.

Em robótica, potenciais artificiais tem sido usado extensivamente para produzir resultados nas leis de controles (RIMON; KODITSCHKEK, 1992; KHATIB, 1986) e desviar de obstáculos. Modelagem potencial também têm sido utilizado para estabilizar sistemas mecânicos de forma eficiente (BLOCH et al., 2001). A utilização de potenciais artificiais em tarefas em grupo como endereçar um problema de robôs autônomos (KODITSCHKEK, 1994) e coordenar constelações de espaçonaves (SCHARF; HADAEGH; PLOEN, 2004) . Na indústria de inteligência artificial e animação, heurísticas similares de regras de tráfego são impostas para reproduzir comportamento coordenado (REYNOLDS, 1987). A figura 3.4 exemplifica as forças agindo sobre veículos f_I , as forças agindo sobre líderes virtuais f_h e a velocidade da formação v_0 .

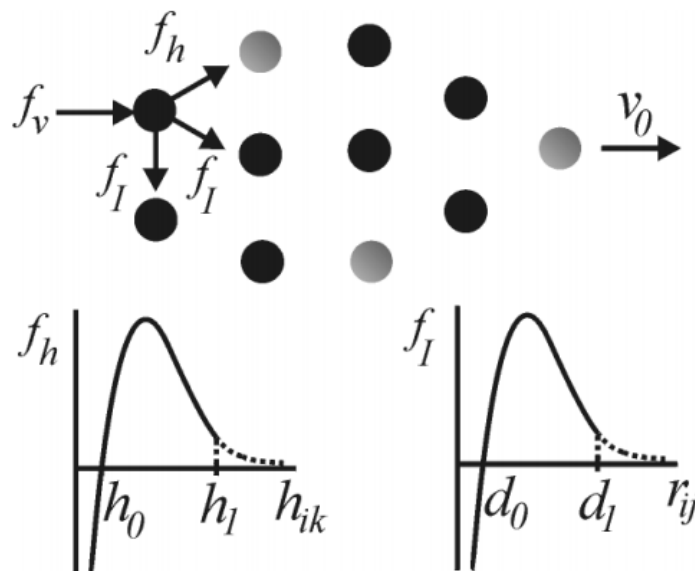


Figura 3.4 – Formação em grupo com forças de controle.

A abordagem de representar líderes virtuais pode ser necessária em situações de desvio de obstáculos, manobras e tomada de decisões. A solução visa criar pontos objetivo adicionais referentes aos objetos vizinhos, assim, podendo ser manipulados para cada cenário. A necessidade do simulador da existência de movimentos em comboio é uma grande motivação para utilizar essa abordagem.

4 ARQUITETURA

O controlador trata as decisões tomadas por um veículo em um ambiente virtual 3D e foi dividido em dois módulos. Como podemos verificar na figura 4.1, o procedimento para começar um movimento inicia quando o usuário deseja mover a(s) viaturas selecionada(s) para uma posição e envia-a para o algoritmo de path-finding. Ao calcular o caminho pelo algoritmo escolhido de path-finding, o itinerário dado por uma lista de posições é enviado para o controlador autônomo para começar o movimento.

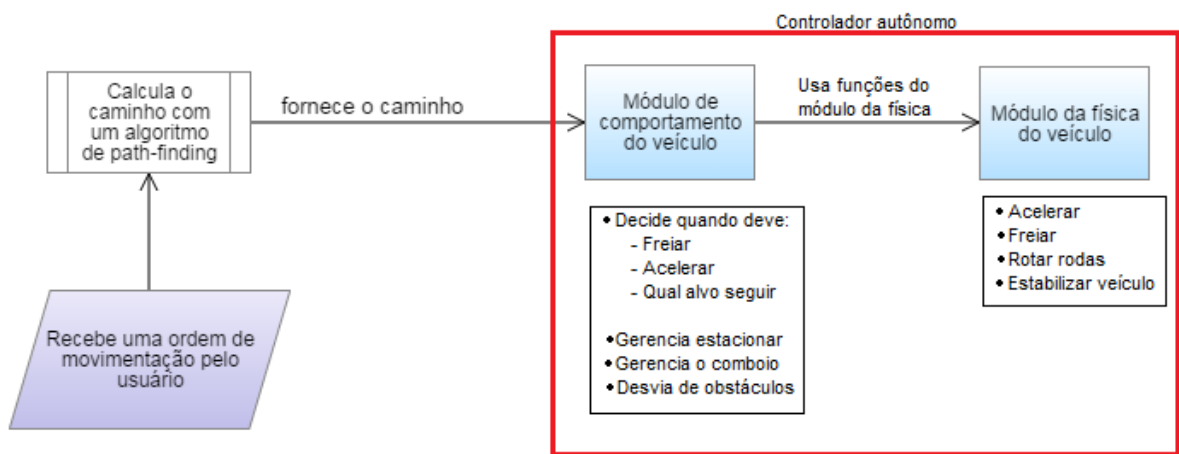


Figura 4.1 – Fluxo de informações do controlador

Os veículos vão se organizar na maneira em que se dispõe na ordem dada pelo usuário e se gerenciar para que naveguem sobre as posições dadas. O módulo de comportamento trata toda a comunicação entre veículos e verifica qual ação tomar pelo automóvel até estacionar. O módulo da física implementa a interface utilizada pelo comportamento que tem a função básica de padronizar a distribuição de forças aplicadas no modelo e estabilizar o veículo em movimento.

5 IMPLEMENTAÇÃO

Neste capítulo é apresentado uma breve descrição do motor gráfico Unity3D no qual foi utilizado para a implementação do controlador e as principais funcionalidades do controlador. Foi organizado em dois módulos distintos, conforme a arquitetura apresentado no capítulo 4, dividida em física e comportamento. O módulo de física implementa mais especificamente as interações físicas do motor do veículo como rotar a roda, aplicar forças, e o módulo de comportamento dita a tomada de decisão do veículo em uma dada situação.

5.1 Unity3D

A Unity3D é um motor gráfico multiplataforma para jogos 3D compatível com WebGL, Windows, MAC e Linux. Possui diversas funções básicas para a simulação de um ambiente virtual como a utilização da biblioteca PhysX(incluindo detector de colisão, soft body e ragdoll), API robusta com grande documentação, suporte para uso de shaders e muitas outras funcionalidades nativas. A importação automatizada de modelos 3D compatível com os programas mais conceituados na área de modelagem é outro aspecto que foi utilizado e que agiliza bastante o processo de geração de um ambiente 3D.

Além de possuir uma interface amigável, a Unity3D oferece duas linguagens diferentes para a implementação do projeto, sendo elas C# ou Javascript(também conhecido como UnityScript pois é uma implementação da sintaxe estilo javascript). Entre todas as funcionalidades do motor, as mais importantes para a confecção deste trabalho foi a biblioteca física, Rigidbody e colisores.

5.1.1 Rigidbody

O Rigidbody é um componente fornecido pelo motor que transfere o controle da movimentação do objeto para o motor de física da Unity3D. Deve-se adicionar colisores para fazer interações de colisão entre o objeto e outros objetos ou terreno.

Utilizando esse componente, funções básicas como gravidade e simulação de forças são manipulados por uma API de forma realista, assim abstraindo a parte do cálculo físico para o motor. O componente têm campos para serem preenchidos de acordo com o objeto que se deseja representar baseando-se no modelo 3D e seus colisores, como massa, centro de massa,

fricção, entre outros.

5.1.2 Colliders

Os colisores são componentes essenciais para qualquer simulação física. Eles expressam os limites de colisão de cada objeto como mostra a figura 5.1, sendo assim, colisores que referenciam rigidbodies diferentes não estão na mesma posição e ao mesmo tempo.

O cálculo de intersecção é a principal característica de colisões, sendo assim, primitivas regulares, como cubos, capsulas, esferas são de complexidade baixa para calcular a intersecção entre os mesmos. Em contraste, colisores convexos mais complexos têm um aumento na complexidade proporcional à quantidade de pontos.

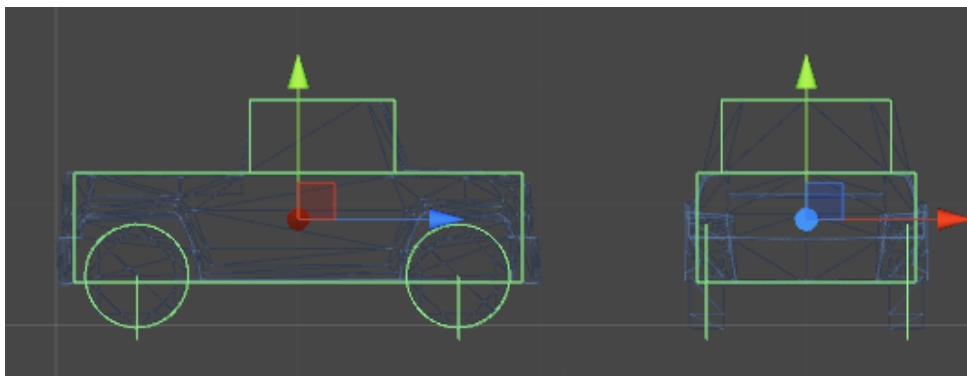


Figura 5.1 – Em verde é representado todos os colisores do veículo

5.1.2.1 Wheel Collider

O colisor utilizado para simular rodas de veículos é o nativo Wheel Collider como representado na figura 5.2. Esse colisor é representado pela distância do centro da roda até seu raio somado com a altura da suspensão e verificado contra outros objetos. Há pontos positivos na escolha desse colisor, como o cálculo de suspensão, tração e fricção baseando na biblioteca física da Unity3D, uma vez que pode-se implementar um colisor de roda mais realista.

5.1.3 Centro de Massa

Na física, o centro de massa de uma distribuição de massa no espaço significa um único ponto onde a posição relativa ao peso da massa distribuída soma zero, ou seja, o ponto onde caso seja aplicada uma força em uma dada direção o objeto irá se mover na direção da força sem sofrer rotação em nenhum eixo. Cálculos em mecânica são normalmente simplificados quando



Figura 5.2 – Imagem de um colisor de roda.

formulados em função do centro de massa. O centro de massa é um ponto hipotético onde a massa total do objeto está concentrada.

No Unity3D há um ponto que representa o centro de massa de um objeto como ilustra a figura 5.3. Diferente da física, esse ponto é representado pela média da posição e tamanho de todos os colisores. Como somente o componente Rigidbody possui controle da física do veículo e conseqüentemente a posição do centro de massa, o mesmo deve ser modificado para representar corretamente o modelo do automóvel. Para veículos, é necessário ter um centro de massa próximo ao chão, sendo assim vem a necessidade de modificá-lo para posições ideais para simular a estabilidade dos automóveis.

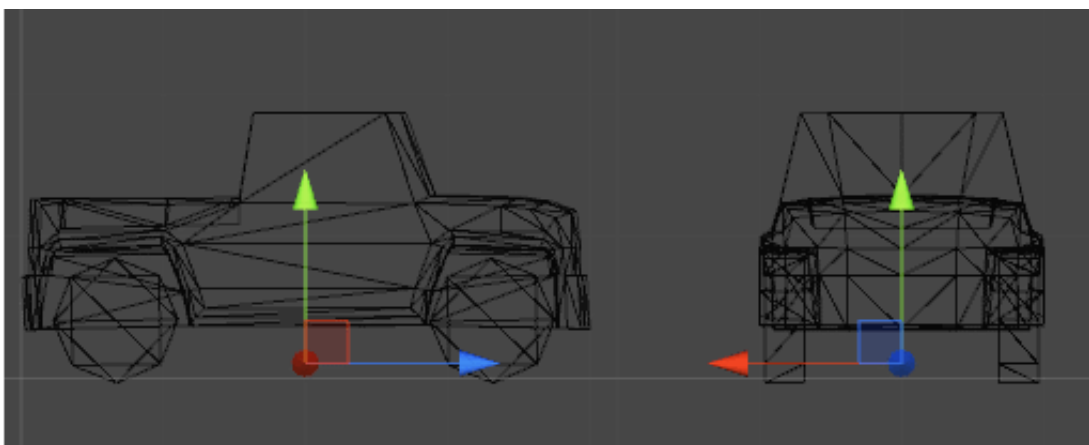


Figura 5.3 – Figura representando o centro de massa atribuído ao modelo.

5.1.4 Módulo da física

Para simular fielmente a física de veículos em um ambiente virtual, deve-se utilizar o mesmo padrão na aplicação de forças sobre suas rodas para uma mesma movimentação, assim, utilizando uma mesma implementação. Todo movimento realizado pelo controlador utiliza as implementações para aplicação de forças descritas nos próximos tópicos para uma maior fidelidade visual e física. Esse módulo implementa as interações mais básicas do veículo com a API física da Unity3D e seus colisores, modelando a aplicação de forças sobre o objeto e a maneira que ele se comporta no ambiente virtual 3D.

5.1.4.1 Forças - Aceleração e Frenagem

Forças são essenciais para qualquer simulação física. Ao aplicar força nas rodas à partir do WheelCollider é possível fazê-lo andar de forma realista, levando em consideração parâmetros de simulação física. A magnitude da força é proporcional à aceleração e depende da massa do veículo, forças de atrito entre o chão e o pneu, potência do motor como também do centro de massa para que se obtenha um movimento sem que o veículo perca estabilidade. A abordagem do steering behaviour utiliza aplicação de forças instantâneas em agentes que podem rotar em seu próprio eixo, diferente de veículos, assim a principal diferença é a tradução do steering behaviour para ser utilizado em veículos realistas.

Da mesma maneira que para se acelerar, a frenagem é feita com uma força no sentido inverso do movimento e aplicada diretamente nas rodas. O módulo implementa dois tipos de frenagem. O primeiro é uma frenagem progressiva, onde a força é incrementada a medida que o tempo passa. Esse é utilizado para fazer frenagens gradativas para simular uma parada suave. O segundo é uma frenagem brusca, onde a força máxima de parada é aplicada nas rodas instantaneamente. Esse tipo é utilizado para frenagens bruscas.

5.1.4.2 Rotação das rodas e Barra estabilizadora

Saber o ângulo de rotação da roda do veículo é uma das funcionalidades do módulo físico e é essencial para mover-se em direção a uma posição objetivo. A função de calcular esse ângulo recebe como parâmetro um ponto no espaço e a partir dele é calculado quanto o eixo dianteiro deve rotacionar para que o agente se movimente naquela direção. Além disso, é calculado o giro horário ou anti-horário da roda, para mover para esquerda ou direita. Após

esses cálculos, é aplicado uma rotação gradativa nas rodas para que o automóvel se direcione para o alvo até um limite máximo de rotação.

Outro fator que deve ser tratado é a estabilidade de veículos em curvas é uma tarefa complexa e realizada por diversas peças especializadas. É possível estabilizar automóveis em ambientes virtuais simulando apenas a principal delas: a barra estabilizadora. As barras estabilizadoras ligam as duas rodas do mesmo eixo permitindo um grau limite de liberdade entre eles como mostra a figura 5.4. Quando uma das rodas é empurrada para cima, a barra transfere uma parte da força para a outra roda, assim sua suspensão também é comprimida. Isso limita a rotação da carroceria do veículo naquele eixo dificultando a capotagem.

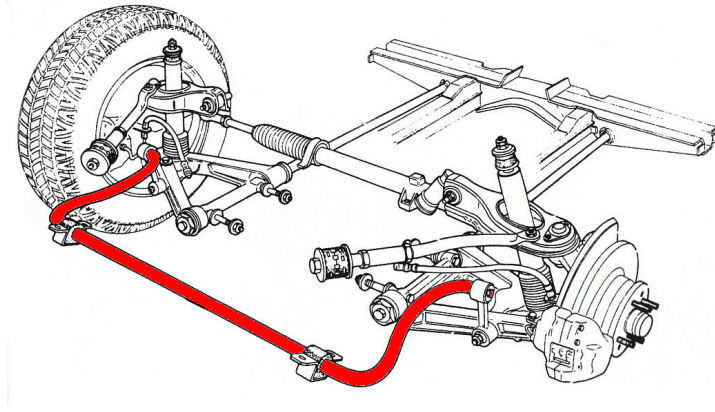


Figura 5.4 – Em vermelho, está destacado a barra estabilizadora.

A implementação de tal funcionalidade visa a transferência de força de uma suspensão para a outra. A quantidade de transferência é calculada por:

```

if ( wheelLIsGrounded )
    travelL = WheelLocalPosition.y / WheelSuspensionDistance;
else
    travelL = 1.0f;

antiRollForce = (travelL - travelR) * AntiRoll;

```

O trecho de código acima calcula a compressão da suspensão das rodas (travelL e travelR). Caso a roda não esteja no chão ela não está comprimida, com valor 1.0f e caso contrário ela poderá assumir um valor mínimo de 0 (totalmente comprimida). A força de transferência é calculada pela diferença de compressão entre as duas suspensões multiplicada pela força máxima que a barra pode transferir.

5.1.4.3 Representação Visual das Forças Aplicadas nas Rodas e Eixos

Em uma simulação virtual que visa reproduzir a realidade, deve-se levar em conta a reprodução visual da física, como girar as rodas nos eixos vertical e horizontal, balanços e movimentos da suspensão. Para isso, utilizou-se os parâmetros oferecidos pelo wheel collider que fornece meios para representar tais efeitos. As rotações verticais e horizontais são lidas do colisor da roda e reproduzidas no modelo para sua representação visual.

Foi implementado dois tipos de suspensão. A primeira sendo uma suspensão de mola independente em cada roda fazendo seu movimento no eixo vertical e sua força é calculada pelo Wheel Collider e ao aplicar no eixo correspondente permite representar sua oscilação. O segundo é a suspensão do tipo Tatra, que o a rotação é feita na base da espinha como ilustra a figura 5.5. Ambas as suspensões são utilizadas nos veículos apresentados no simulador.

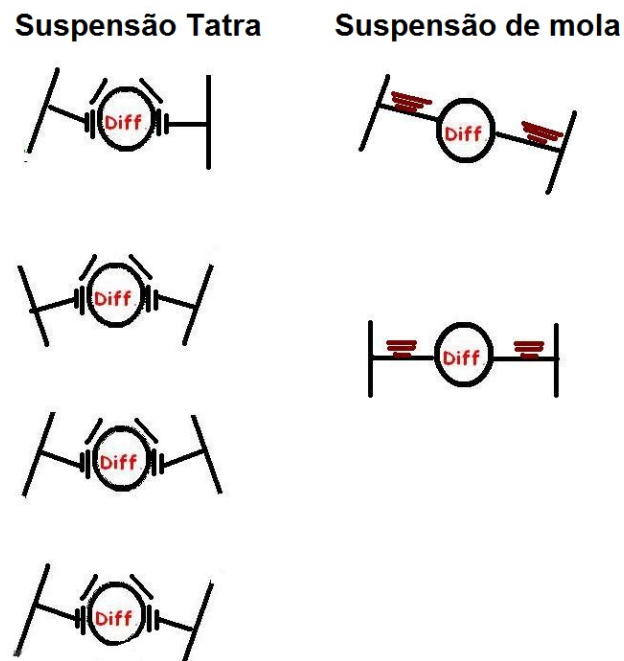


Figura 5.5 – Diferença entre as suspensões.

5.2 Módulo do comportamento

O comportamento do veículo deve assemelhar-se a escolhas que seriam realizadas por um ser humano, em uma dada situação. No contexto do Simulador SIS-ASTROS, foram destacadas as necessidades de locomoção dado um caminho definido por um conjunto de pontos. É importante ressaltar que o problema do simulador se encontra no deslocamento de um con-

junto de viaturas em comboio ou de forma individual, com caminhos iguais ou não, mas sempre seguindo o itinerário.

Algumas alterações precisam ser feitas para que a solução do steering behaviour funcione em veículos. Como foi explicado, o steering behaviour funciona como um somatório de forças, porém, veículos não podem sofrer rotações instantâneas. As rotações em veículos devem ser gradativas e são necessariamente aplicadas em suas rodas e não em sua carroceria. Para tratar essa situação, deve-se aumentar a distância de detecção de bloqueio de passagem baseando-se na sua velocidade e na rotação máxima de suas rodas, fazendo com que o veículo enxergue obstáculos mais distantes possibilitando a manobra.

As dimensões de agentes pequenos podem ser simplificadas com somente um raio de teste, porém, como veículos têm dimensões maiores ocorre a necessidade de realizar testes com um colisor na frente do veículo, proporcional ao seu tamanho, para se detectar com maior precisão a necessidade de um desvio evidenciado na figura 5.6. Além disso, é importante analisar além de uma linha na dianteira, é necessário um retângulo na parte dianteira do veículo mais uma margem de segurança proporcional a largura da viatura. Essa necessidade surge caso os objetos de desvio sejam dinâmicos, como outros veículos, e estejam em uma rota de colisão. Nesse caso deve-se prever o movimento dos agentes, verificando e evitando possíveis colisões.

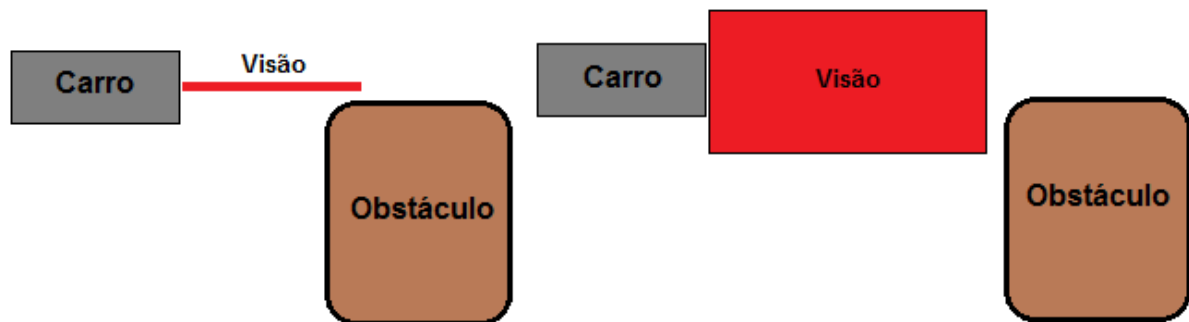


Figura 5.6 – Diferença entre visão com raio e com um retângulo.

5.2.1 Controle de velocidade atual

O controle de velocidade é dado por dois limites, um estipulado pelo motor do veículo, a velocidade máxima que o motor suporta, e outro estipulado pelo usuário. Enquanto a velocidade atual for menor que um dos dois limites, o veículo irá acelerar até alcançar o menor deles, respeitando o limite de velocidade imposto pelo usuário, pelo motor e pelo ambiente no qual o automóvel se encontra. A aceleração é dada pela força de tração aplicada nas rodas do veículo,

e é incrementada a medida que o veículo acelera, até um limite superior estipulado pelo usuário.

Há a necessidade de controlar a velocidade atual ao seguir outro veículo, pois não pode haver uma diferença grande entre as duas. Também deve-se manter uma distância mínima de segurança de um automóvel para o outro. A distância de segurança se traduz no espaço onde a viatura conseguirá frear sem que ocorra uma colisão no veículo da frente. Então deve-se verificar a quantidade de aceleração e frenagem a cada momento da simulação.

O estacionar das viaturas é tratado quando o veículo atinge o último ponto do itinerário e quando o veículo no qual está seguindo está estacionado em seu ponto final. A velocidade é reduzida a medida que se aproxima do ponto de parada calculado pela distância do veículo da frente ou o fim do caminho.

5.2.2 Seguir Fielmente o Caminho

Dado um caminho pré-computado por um algoritmo de path-finding escolhido, o veículo deve seguir fielmente o caminho, passando por todos os pontos. Ao chegar em uma posição destino, deve-se verificar a existência de mais posições dentro do itinerário e trocar o alvo assim que uma distância mínima entre o segmento de reta sendo analisado for atingida. Caso seja a última posição do itinerário, o agente chegou ao destino final.

Para se deslocar para um alvo, é necessário achar um ângulo de rotação das rodas para que o veículo se dirija para a localização destino. O ângulo é calculado da seguinte forma:

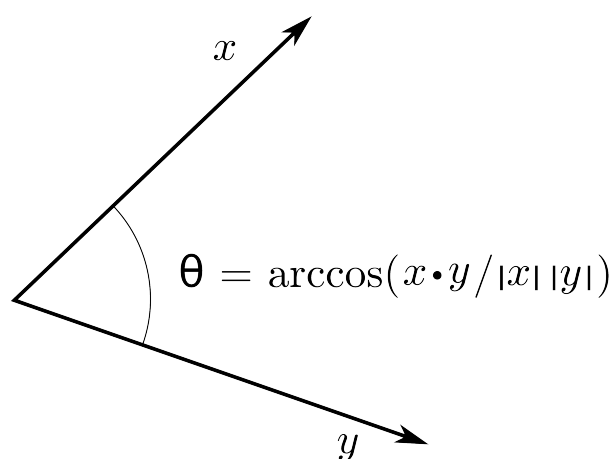


Figura 5.7 – Cálculo de ângulo entre dois vetores

De forma simplificada, encontrando as direções forward do veículo e sua direção para o alvo, o ângulo entre dois vetores é calculado como mostra a figura 5.7. Porém somente o cálculo do ângulo não é suficiente para girar até o destino, como explicado na sessão 5.1.4.2

referente à rotação das rodas.

5.2.3 Deslocamento em Comboio

Para se deslocar em comboio é necessário ter uma comunicação com os outros veículos para que suas situações atuais convirjam em um movimento em conjunto (FAX; MURRAY, 2004). Foi dividido em dois estágios o deslocamento em comboio, o primeiro é quando nenhum veículo chegou ao seu destino, e o segundo quando já há veículos estacionados. A primeira parte desse comportamento o veículo deve seguir o itinerário dado, pois uma das necessidades do simulador é que os veículos que se deslocam em comboio devem seguir exatamente o mesmo caminho, respeitando o veículo sucessor. Já a segunda, o automóvel deve comparar os itinerários e verificar seu sucessor e analisar se deve estacionar atrás do veículo à seguir ou desviar e continuar o deslocamento.

Na hora de definir um itinerário pode-se ordenar as viaturas dentro de um comboio, vide figura 5.8. Assim, desde o começo da simulação, é enviado os veículos selecionados para realizar o movimento na ordem correta para atribuição ao comboio e seus itinerário. A ordem dos veículos dentro do movimento em grupo é retirada da disposição em que se encontram na lista selecionada na definição do itinerário.

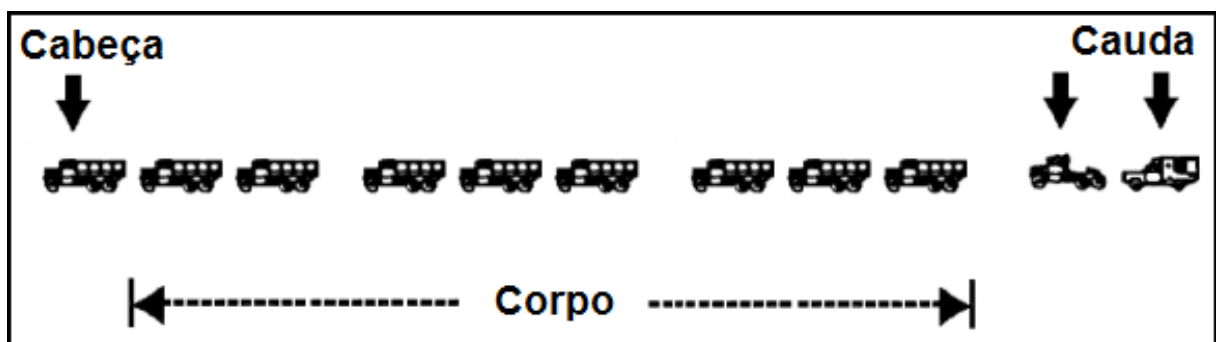


Figura 5.8 – Deslocamento em comboio.

5.2.4 Garantir a ordem das viaturas dentro do comboio

Para se garantir a ordem das viaturas dentro de um comboio, é necessário saber em qual parcela do itinerário se encontra o próprio agente e o veículo no qual deve seguir. Essa informação é necessária para verificar se a ordem dos veículos está de acordo com o requisitado. Para averiguar a ordenação das viaturas, deve-se ter meios de comparar suas posições dentro do itinerário. Caso apresente necessidade de espera pela viatura alvo, o agente deve parar e aguardar que seu alvo o ultrapasse para continuar o deslocamento.

A comparação pode ser feita por meio do índice da lista de posições dentro do itinerário e, caso estejam no mesmo índice, comparar a porcentagem já percorrida. Segue o cálculo:

```
pathLength = ( itinerary [ i - 1 ] - itinerary [ i ] ). magnitude ;
itineraryProgress = distFromLastPoint / pathLength ;
```

Ao se calcular a distância do último ponto do percurso atribuindo em `distFromLastPoint` e calculando `pathLength` como sendo a distância do sub-caminho, basta fazer uma divisão do primeiro pelo segundo que resultará na porcentagem que foi percorrida pelo veículo, dentro do sub-caminho. Assim, é possível comparar o progresso sobre do itinerário.

5.2.5 Comportamento de Redirecionamento

O comportamento de redirecionamento é uma funcionalidade necessária para o simulador. A escolha de implementação foi uma aproximação da solução *obstacle avoidance* de Reynolds (REYNOLDS, 1999) com algumas modificações para a Unity3D e o modelo utilizado no simulador.

5.2.5.1 Sensores

A verificação de colisão continua utilizando a aproximação de uma circunferência para os obstáculos. É utilizado um sensor na frente da viatura para detectar a colisão com obstáculos e verificar o mais imediato para efetuar o desvio.

A implementação utilizou detecção de colisão de retângulo e circunferência em vez de linha-circunferência para dar mais robustez ao desvio, uma vez que há a necessidade de reproduzir a largura do agente, como pode ser visualizado na figura 5.9. Se a distância for menor ou igual ao raio da esfera, então o vetor está dentro da esfera e uma colisão foi encontrada.

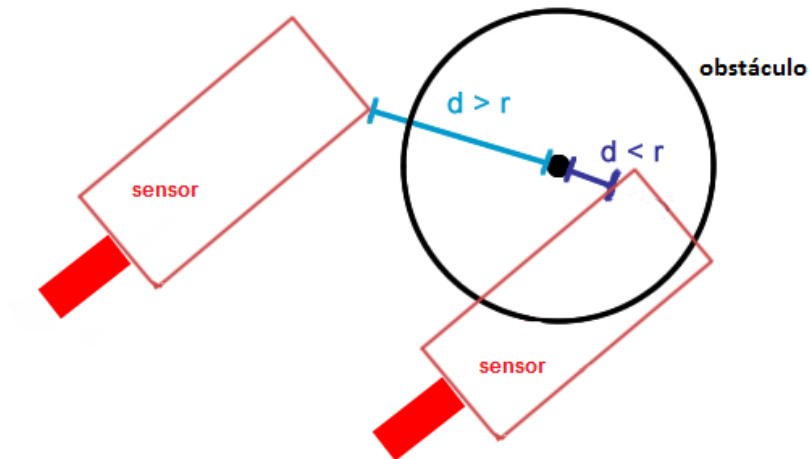


Figura 5.9 – Verificando colisão

5.2.5.2 Variação das Forças do Steering Behaviour para Utilização em Veículos

Algumas modificações precisam ser efetuadas para que a solução do desvio de obstáculos do steering behaviour funcione em veículos. Como foi explicado, o steering behaviour funciona como um somatório de forças, porém, veículos não podem sofrer rotações instantâneas. As rotações em veículos devem ser gradativas e são necessariamente aplicadas em suas rodas e não em sua carroceria. Para corrigir essa situação, deve-se aumentar a distância de detecção de bloqueio de passagem baseando-se na sua velocidade e na rotação de direcionamento máxima de suas rodas.

As dimensões de agentes pequenos podem ser simplificadas com somente um raio de teste, porém, como o veículo tem dimensões maiores ocorre a necessidade de realizar testes com um colisor na frente do veículo, proporcional ao seu tamanho, para se detectar com maior precisão a necessidade de um desvio. Além disso, é importante analisar além de somente a dianteira, mas também um retângulo na parte dianteira do veículo. Essa necessidade surge uma vez que há objetos de desvio dinâmicos, como outros veículos, e atuam em uma rota de colisão.

Nesse caso deve-se prever o movimento dos agentes, verificando e evitando possíveis colisões.

6 RESULTADOS

Este trabalho baseou-se na necessidade do simular veículos autônomos em um ambiente virtual que realizem as atividades apresentadas pelo simulador. Não foi encontrada nenhuma ferramenta que forneça soluções para os cenários apresentados no simulador, assim, foi implementado e modelado o controlador de forma que cada situação tenha sua solução.

O produto final é um sistema que controla o deslocamento de veículos genéricos em um ambiente virtual 3D, ou seja, pode ser simulado qualquer modelo de automóvel. Os caminhos devem ser enviados para o controlador para começar o movimento. A figura 6.1 mostra um caminho gerado representado pelas esferas.

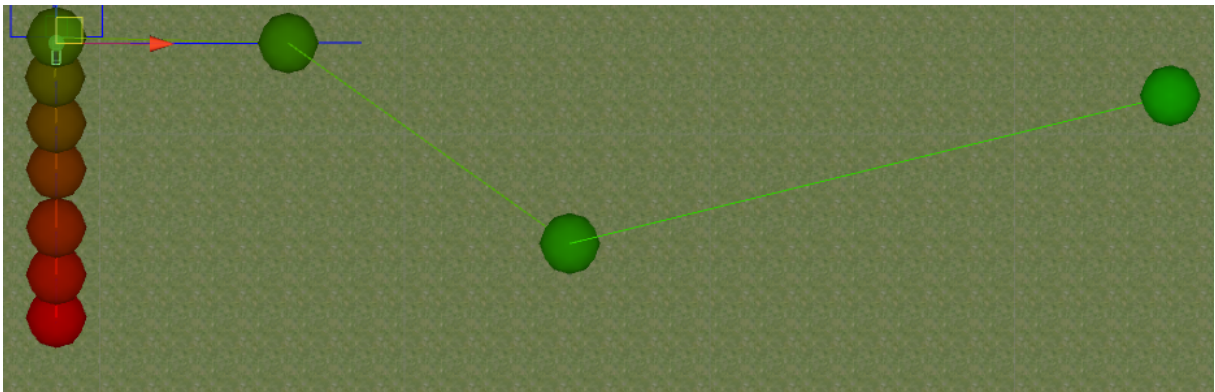


Figura 6.1 – Caminho que será percorrido pelas viaturas.

Além de percorrer um caminho dentro do ambiente, pode haver obstáculos entre o veículo e a posição destino e nesses casos ocorrerá o desvio de tais objetos. Tal decisão será tomada quando obstáculos colidirem com os sensores. O cálculo da força necessária depende da distância do mesmo. A figura 6.2 ilustra uma viatura desviando de um obstáculo representado pela esfera branca, um retângulo azul representando o sensor e um retângulo verde representando o sensor de obstáculos imediatos para desvio. As linhas que partem da esfera e da viatura representam as forças de repulsão e força resultante respectivamente. A esfera vermelha representa as posições do caminho e a linha em vermelho partindo da mesma representa a direção do próximo destino.

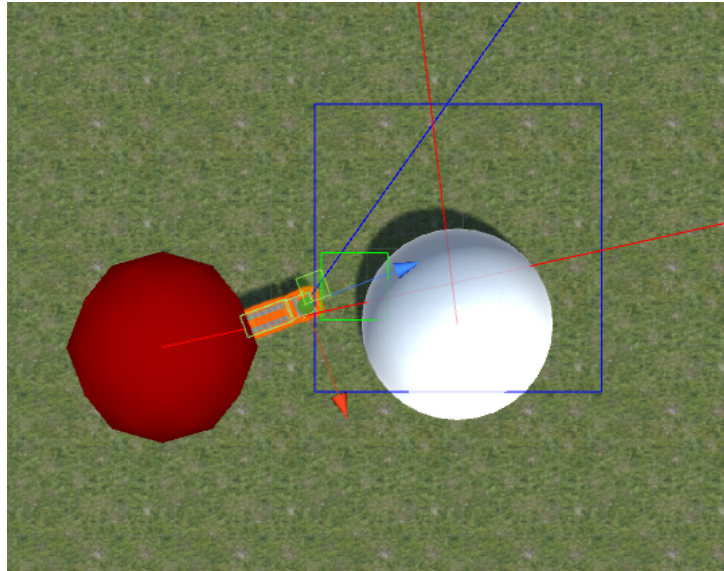


Figura 6.2 – Essa imagem ilustra o steering behaviour implementado dentro do motor gráfico.

O controlador oferece suporte para o deslocamento em grupo de viaturas. Esse movimento usa um mesmo caminho para todas as viaturas e garante um espaçamento entre as mesmas, como visto na figura 6.3. Caso as viaturas estejam em ordem diferente da requisitada, o comboio irá se organizar de tal forma para alcançar a ordenação desejada.

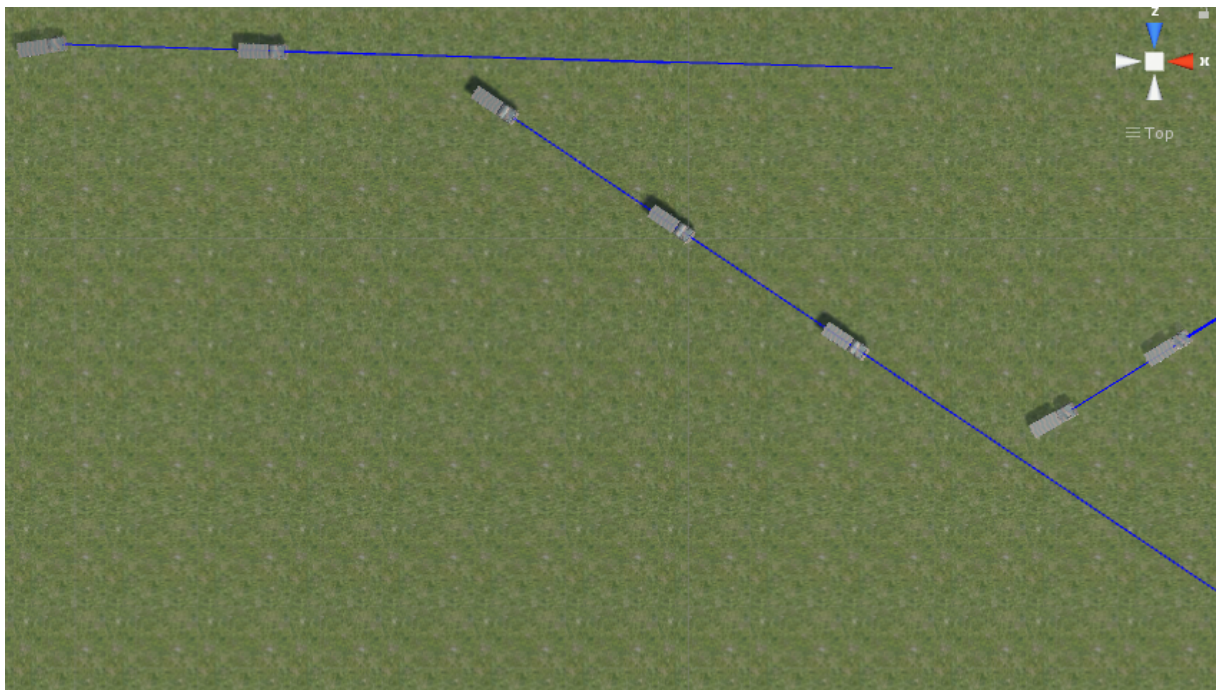


Figura 6.3 – Deslocamento em comboio de viaturas por um mesmo caminho.

Para finalizar um deslocamento, um veículo deve chegar à sua posição destino e parar, como ilustra a figura 6.4. Essa posição é definida pelo itinerário e pelo comboio. Para verificar

se um veículo chegou ao final do itinerário, basta examinar a projeção do último segmento de reta do caminho com o ponto final da trajetória e caso seja igual, estacionar. Para estacionar uma viatura em um deslocamento em comboio, ocorre a comunicação entre os veículos para se verificar a necessidade de parada, analisando distância e itinerários.

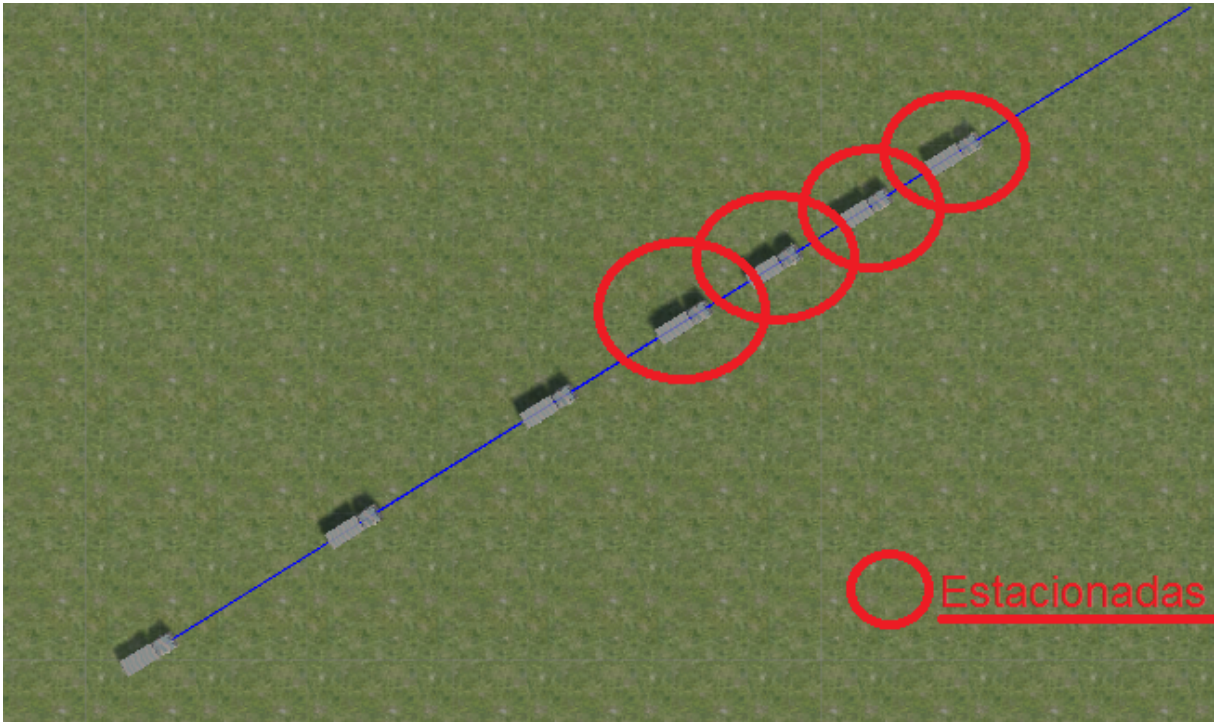


Figura 6.4 – Figura ilustrando viaturas estacionando em comboio.

7 CONCLUSÃO

O objetivo deste trabalho foi a implementação de um sistema de controle autônomo de veículos para ambientes virtuais. Foi feita uma aproximação de controladores de agentes humanos para sua utilização em veículos auto-motores além de agregar características essenciais para o simulador Astros2020.

A meta de elaborar um controlador de veículos genéricos foi alcançada uma vez que é possível reproduzir o comportamento de veículos utilizando esta proposta. Pode ser utilizado em qualquer ambiente onde há a existência de automóveis e a necessidade de reproduzir suas movimentações de forma robusta. O controlador oferece a tomada de decisão de veículos em qualquer ambiente virtual e a reprodução da física. As decisões tomadas são controle de velocidade, seguir um caminho, deslocamento em grupo, desvio de obstáculos. A física é reproduzida com colisores, aceleração, frenagem e centro de massa.

Dado um caminho pré-computado por um algoritmo de path-finding o controlador deve garantir que o agente alcance o destino. Para isso, o controlador se regula a situação imediata dos arredores do automóvel para mudar seu comportamento.

Os resultados obtidos mostram-se satisfatórios uma vez que testes exaustivos são executados no simulador, o controlador encontrou e alcançou o destino em todas as execuções. Este trabalho traz consigo uma solução para a movimentação em grupo de veículos possibilitando a reprodução dos mesmos em ambientes virtuais com diversos possíveis incrementos como realizar um planejamento para desvio de obstáculos, comportamento em rodovias, estradas, análise de terreno para possíveis ajustes na velocidade, entre outros.

REFERÊNCIAS

- BLOCH, A. M. et al. Controlled Lagrangians and the stabilization of mechanical systems. II. Potential shaping. **IEEE Transactions on Automatic Control**, [S.l.], v.46, n.10, p.1556–1571, 2001.
- CHOSSET, H. M. **Principles of robot motion: theory, algorithms, and implementation**. [S.l.]: MIT press, 2005.
- EGERSTEDT, M.; HU, X.; STOTSKY, A. Control of mobile platforms using a virtual vehicle approach. **IEEE Transactions on Automatic Control**, [S.l.], v.46, n.11, p.1777–1782, 2001.
- FAX, J. A.; MURRAY, R. M. Information flow and cooperative control of vehicle formations. **IEEE**, [S.l.], v.49, n.9, p.1465–1476, 2004.
- KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **The international journal of robotics research**, [S.l.], v.5, n.1, p.90–98, 1986.
- KODITSCHKEK, D. E. An approach to autonomous robot assembly. **Robotica**, [S.l.], v.12, n.02, p.137–155, 1994.
- LATOMBE, J.-C. **Robot motion planning**. [S.l.]: Springer Science & Business Media, 2012. v.124.
- LEONARD, N. E.; FIORELLI, E. Virtual leaders, artificial potentials and coordinated control of groups. In: DECISION AND CONTROL, 2001. PROCEEDINGS OF THE 40TH IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2001. v.3, p.2968–2973.
- REYNOLDS, C. W. Flocks, herds and schools: a distributed behavioral model. **ACM SIGGRAPH computer graphics**, [S.l.], v.21, n.4, p.25–34, 1987.
- REYNOLDS, C. W. Steering behaviors for autonomous characters. In: GAME DEVELOPERS CONFERENCE. **Anais...** [S.l.: s.n.], 1999. v.1999, p.763–782.
- RIMON, E.; KODITSCHKEK, D. E. Exact robot navigation using artificial potential functions. **IEEE Transactions on robotics and automation**, [S.l.], v.8, n.5, p.501–518, 1992.

SCHARF, D. P.; HADAEGH, F. Y.; PLOEN, S. R. A survey of spacecraft formation flying guidance and control. part ii: control. In: AMERICAN CONTROL CONFERENCE, 2004. PROCEEDINGS OF THE 2004. **Anais...** [S.l.: s.n.], 2004. v.4, p.2976–2985.

SMITH, T. R.; HANSSMANN, H.; LEONARD, N. E. Orientation control of multiple underwater vehicles with symmetry-breaking potentials. In: DECISION AND CONTROL, 2001. PROCEEDINGS OF THE 40TH IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2001. v.5, p.4598–4603.

SUD, A. et al. Real-time path planning for virtual agents in dynamic environments. In: ACM SIGGRAPH 2008. **Anais...** [S.l.: s.n.], 2008. p.55.

YANAKIEV, D.; KANELLAKOPOULOS, I. A simplified framework for string stability analysis in AHS. In: IFAC WORLD CONGRESS, 13. **Proceedings...** [S.l.: s.n.], 1996. v.182, p.177–182.