

Guilherme Bezerra Zamprônio

**Um Sistema de Simulação 3D de Evacuação de Emergência
em Plataformas de Petróleo**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador: Prof. Alberto Barbosa Raposo
Coorientador: Prof. Marcelo Gattass

Rio de Janeiro
Março de 2015

Guilherme Bezerra Zampronio

**Um Sistema de Simulação 3D de
Evacuação de Emergência em Plataformas
de Petróleo**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Alberto Barbosa Raposo

Orientador e Presidente
Departamento de Informática – PUC-Rio

Prof. Marcelo Gattass

Coorientador
Departamento de Informática – PUC-Rio

Prof. Waldemar Celes Filho

Departamento de Informática – PUC-Rio

Felipe Gomes de Carvalho

Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro
Departamento de Informática – PUC-Rio

Rio de Janeiro, 02 de março de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Guilherme Bezerra Zampronio

Graduou-se em Informática pela Pontifícia Universidade Católica do Rio de Janeiro em 2007. Desde 2008, trabalha na Petrobras como analista de negócio para sistemas de Segurança, Meio Ambiente, Eficiência Energética e Saúde (SMES).

Ficha Catalográfica

Zampronio, Guilherme Bezerra

Um sistema de simulação 3D de evacuação de emergência em plataformas de petróleo / Guilherme Bezerra Zampronio; orientador: Alberto Barbosa Raposo; co-orientador: Marcelo Gattass. – 2015.

57 f. : il. (color.) ; 29,7 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2015.

Inclui bibliografia

1. Informática – Teses. 2. Simulação de emergência. 3. Multidão. 4. Engine de jogos. 5. 3D. 6. Petróleo e gás. 7. Agentes. 8. Comportamento. I. Raposo, Alberto Barbosa. II. Gattass, Marcelo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Agradecimentos

Ao meu orientador Professor Alberto Barbosa Raposo e coorientador Professor Marcelo Gattass pelos ensinamentos, dedicação e parceria para a realização deste trabalho.

À Petrobras pelos auxílios concebidos, sem os quais este trabalho não poderia ter sido realizado.

Aos funcionários da Petrobras.

A minha companheira Jaqueline, por todo incentivo e paciência durante todos esses anos.

Aos meus pais e irmão, pela educação, apoio e carinho.

À PUC-Rio pela estrutura e auxílios, oferecendo condições para o desenvolvimento deste trabalho.

Aos professores que participaram da Comissão examinadora.

A todos os professores e funcionários do Departamento de Informática pelos ensinamentos e ajuda.

Aos meus colegas da PUC-Rio

Ao Tecgraf e todos os seus funcionários e pesquisadores.

Resumo

Zamprônio, Guilherme Bezerra; Raposo, Alberto Barbosa. **Um Sistema de Simulação 3D de Evacuação de Emergência em Plataformas de Petróleo**. Rio de Janeiro, 2015. 57p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Uma aplicação de simulação de evacuações, usando recursos computacionais, pode ajudar a prever situações, fluxos, conflitos e comportamentos que somente acontecerão em uma real situação de perigo. Este tipo de aplicação possibilita a execução de inúmeros cenários diferentes a qualquer momento, desde que previamente especificados e desenvolvidos, sem a custosa e complexa alocação de pessoas reais. Esta dissertação propõe um sistema de simulações de emergência em plataformas de petróleo em 3D com resultados em tempo real utilizando como arquitetura uma engine de jogos (Unity). A solução desenvolvida foi testada em modelos de plataformas reais para comparação com tempos obtidos em simulados de emergência realizados com pessoas. O desempenho do sistema será exposto, assim como futuros trabalhos e melhorias a serem realizadas.

Palavras-chave

Simulação de Emergência; Multidão; Engine de Jogos; 3D; Petróleo e Gás; Agentes; Comportamento.

Abstract

Zamprônio, Guilherme Bezerra; Raposo, Alberto Barbosa (Advisor). **A 3D Simulation System for Emergency Evacuation in Offshore Platforms**. Rio de Janeiro, 2015. 57p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

An application for evacuation simulation using computational resources may help previewing situations, flows, conflicts, and behaviors that may only happen in a real danger situation. This kind of application enables the execution of several pre-defined scenarios at any time, without the expensive and complex allocation of real people. This dissertation proposes an emergency simulation system on oil platforms in 3D with real-time results using as architecture a game engine (Unity). The solution developed was tested in of real platforms models for comparison with times obtained in emergency simulations with people. System performance will be exposed, as well as future works.

Keywords

Emergency Simulation; Crowd; Game Engine; 3D; Oil and Gas; Agents; Behavior.

Sumário

1	Introdução	10
2	Trabalhos relacionados	13
2.1.	Soluções comerciais	13
2.2.	Pesquisas acadêmicas	15
3	Método desenvolvido	22
3.1.	Plataforma de desenvolvimento	22
3.2.	Modelagem	23
3.3.	Desenvolvimento	27
3.3.1.	Câmeras	27
3.3.2.	Agentes	29
3.3.3.	Controle	34
4	Resultados	39
4.1.	Cénario simulado	39
4.2.	Simulação contra a realidade	42
4.3.	Desempenho do software	46
4.4.	Benefícios, melhorias e gargalos	49
5	Conclusão e trabalhos futuros	52
6	Referências bibliográficas	55

Lista de Figuras

Figura 1 - Cena do O Retorno do Rei (2003) criado com o MASSIVE	13
Figura 2 - Tela do software MASSIVE [1].....	14
Figura 3 - Simulação gerada pelo MassMotion [2]	15
Figura 4 - Comportamentos de boids [8].....	17
Figura 5 - CrowdSim executado no Engenhão [16].....	20
Figura 6 - Resultado do CrowdSim [16]	20
Figura 7 - Plataforma do tipo FPSO em forma de navio.....	24
Figura 8 - Configuração da malha de navegação.....	24
Figura 9 - Malha de navegação de uma plataforma semissubmersível ...	25
Figura 10 - Malha de navegação de um FPSO	25
Figura 11 - Configuração dos agentes	26
Figura 12 - Agente renderizado.....	27
Figura 13 - Visão em primeira pessoa.....	28
Figura 14 - Visão de vigilância com 4 câmeras.....	28
Figura 15 - Estados em uma emergência	32
Figura 16 - Rotas de fuga (em vermelho) dos agentes	37
Figura 17 - Painel de controle da cena.....	37
Figura 18 - Três modelos de plataformas utilizados.....	40
Figura 19 - Tempo médio de evacuação.....	43
Figura 20 - Agentes correndo.....	45
Figura 21 - Filas e aglomerados.....	46
Figura 22 - Fim da simulação.....	48

Lista de Tabelas

Tabela 1 - Função Awake.....	33
Tabela 2 - Função Start.....	33
Tabela 3 - Função Update.....	34
Tabela 4 - Função que define o local seguro	35
Tabela 5 - Função que verifica o pathfinding.....	36
Tabela 6 - Configuração dos cenários simulados.....	39
Tabela 7 - Tempo médio de evacuação (ordenada), em minutos	42
Tabela 8 - Tempo médio de evacuação (desordenada), em minutos	44
Tabela 9 - Frames por segundo (FPS) por câmera.....	47
Tabela 10 - Tempo médio de pré-processamento em segundos	48

1 Introdução

Segurança em primeiro lugar. Essa frase nem sempre é considerada na prática pelas pessoas. O que vemos, na maioria das vezes, é um descaso com as normas de segurança. Ninguém pensa que o pior irá acontecer consigo.

O incêndio da boate Kiss¹ em Santa Maria, Rio Grande do Sul, 27 de Janeiro de 2013, despertou novamente a população e autoridades a se preocuparem com medidas de segurança. Neste caso em particular, 242 pessoas morreram e outras 123 ficaram feridas, sendo a segunda maior tragédia em número de vítimas em um incêndio no Brasil.

Nas semanas seguintes, diversas vistorias foram feitas em bares e boates para verificação de saídas de emergência e se as rotas atuais seriam suficientes para uma fuga de todos os presentes nos estabelecimentos em um tempo hábil. Vários lugares foram fechados, sendo solicitados desde novas portas até escadas que suportem um fluxo maior de pessoas. No fundo, foi solicitada obediência e adequação às normas já existentes dos bombeiros.

Em ambientes empresariais, as regras tendem a ser cumpridas mais frequentemente. Nesses casos, além da preocupação com a estrutura do prédio, deve-se ter atenção com o design de salas e estações de trabalho. Não adianta termos escadas e saídas de emergência suficientes, se um escritório está muito cheio e a circulação dentro dele é difícil sem a vazão necessária para evacuar o ambiente em um determinado tempo limite.

É necessário sempre pensar na evacuação destes locais e, para isso, um sistema de apoio é útil nos projetos dos mesmos, evitando a construção de algo que não atende aos requisitos mínimos de segurança. Desta forma, é possível a

¹ Fontes:

<http://g1.globo.com/rs/rio-grande-do-sul/tragedia-incendio-boate-santa-maria/platb/>

http://pt.wikipedia.org/wiki/Incêndio_na_boate_Kiss

verificação de gargalos, possibilitando a criação de rotas alternativas atendendo o fluxo total de pessoas presentes no local em um tempo de evacuação adequado.

Sistemas de simulação são importantes na mitigação e no apoio ao problema. Podem ser utilizados durante todo o processo de vida de um projeto de engenharia. Na elaboração da planta do prédio ou empreendimento qualquer, na validação da construção, em evoluções e manutenções arquiteturas e na conscientização, informação e treinamento dos empregados ou visitantes.

Simulados de emergência com pessoas em ambientes reais e em situações fictícias podem agregar um valor maior à rotina de prevenção a acidentes das empresas. Apesar de serem essenciais para instruir uma força de trabalho, eles sabem que aquela situação não é real e muitas vezes não há o empenho dos mesmos. É custosa a execução de diversos cenários em um curto período de tempo, pois depende do comprometimento de diversos recursos diferentes. Em ambientes grandes como, por exemplo, estádios de futebol, é extremamente difícil alocar a quantidade de pessoas necessárias para executar um simulado.

Uma aplicação de simulação de evacuações, usando recursos computacionais, ajuda a tentar prever situações, fluxos, conflitos e comportamentos que somente acontecerão numa real situação de perigo. Possibilita a execução de inúmeros cenários diferentes a qualquer momento, desde que previamente especificados e desenvolvidos, sem a alocação de pessoas reais.

Porém, representar uma multidão é extremamente difícil e precisa de certos cuidados. Toda multidão é formada por indivíduos independentes com vontades e características próprias. Em ocasiões de risco, que podem ser as mais variadas possíveis, essas características podem se suprimir ou exceder. Além dessas características individuais, temos que retratar o coletivo, onde a necessidade de muitos supera a necessidade de poucos.

Existem diversas possibilidades e vertentes de pesquisa em relação ao tratamento às multidões. Dependendo do escopo do cenário a ser simulado, da quantidade de agentes envolvidos, da qualidade gráfica pretendida e da performance da aplicação, certas técnicas de desenvolvimento podem ser privilegiadas em demérito de outras.

O principal objetivo desta dissertação é implementar um sistema de simulação de evacuação de pessoas em uma situação de emergência em plataformas de petróleo com qualidade gráfica e resultados próximos a realidade.

Este documento está organizado da seguinte forma. O Capítulo 2 discute os trabalhos relacionados, apresentando aplicações comerciais e corporativas e pesquisas acadêmicas. O Capítulo 3 apresenta o método de desenvolvimento escolhido, a arquitetura, configuração, modelagem e implementação da aplicação. Os resultados e um comparativo com a realidade e com os trabalhos relacionados são descritos no Capítulo 4. Finalmente, o Capítulo 5 apresenta as conclusões e trabalhos futuros.

2 Trabalhos relacionados

Existem diversas frentes no mercado atualmente focadas em simulações de multidões e seus comportamentos. Neste capítulo, são apresentadas soluções comerciais, pesquisas acadêmicas com diversos focos em multidões e projetos empresariais.

2.1. Soluções comerciais

Entre as soluções comerciais para simulação de multidões, existem as focadas na indústria de entretenimento (jogos, televisão e cinema) e as focadas em planejamento urbano.

Criado para ser utilizado na trilogia O Senhor dos Anéis (2001), o software MASSIVE [1] virou referência na indústria de entretenimento (Figuras 1 e 2). Ele surgiu da necessidade de representar diversas batalhas entre centenas de milhares de soldados, problema até então não resolvido pela indústria cinematográfica.



Figura 1 - Cena do O Retorno do Rei (2003) criado com o MASSIVE

Utilizando princípios de lógica Fuzzy, os agentes conseguem reagir e interagir com o ambiente e com outros agentes tomando decisões em tempo de execução. Essas reações alteram o seu comportamento na cena de forma realística. Entre as qualidades emotivas simuladas estão: bravura, cansaço, fadiga e alegria.

Cenas de guerras e batalhas podem ser totalmente configuradas e o seu resultado gerado através dessas variações de comportamento.

Modelos mais simples podem ser executados em tempo real enquanto cenas mais complexas são rodadas em grupos incrementais. O seu uso pode ser destacado em filmes tais como: Eu, Robô (2004), King Kong (2005), O Cavaleiro das Trevas (2008) e Guerra Mundial Z (2013).

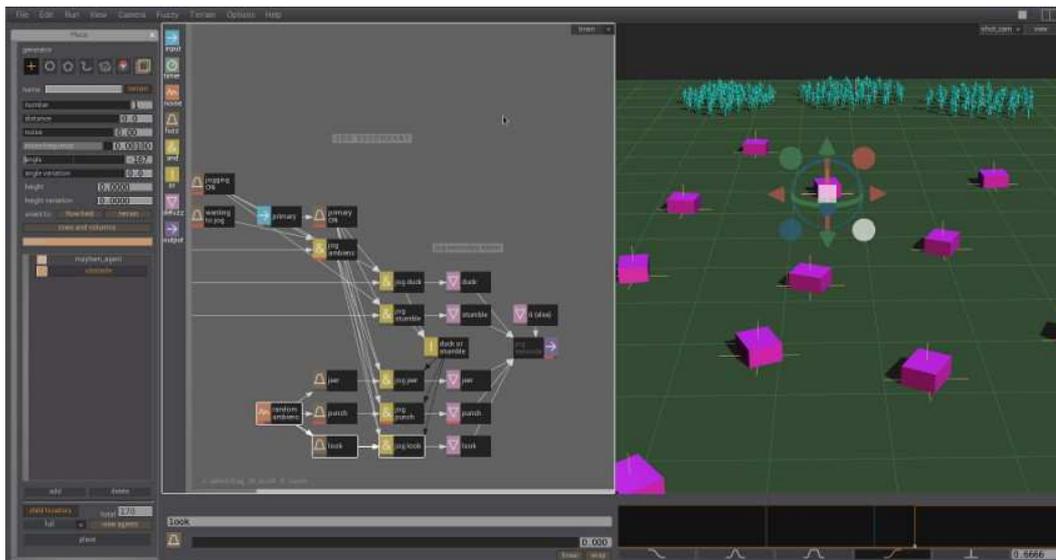


Figura 2 - Tela do software MASSIVE [1]

Outro software comercial de sucesso, porém focado em planejamento urbano é o MassMotion [2]. Tem como seus principais usos, uma avançada simulação de pedestres e análise de multidão. Simula um grande número de pessoas circulando ao mesmo tempo em direções e destinos diferentes, conseguindo uma análise de fluxo por escadas, escadas rolantes, portas e elevadores (Figura 3).

Apesar de obter resultados com ótima qualidade gráfica, o software não executa a simulação em tempo real e não possui variações de comportamento. Foi utilizado em projetos importantes, tais como: Montreal-Pierre Elliot Trudeau International Airport, San Francisco Transbay Terminal e Toronto Union Station.

A Union Station em Toronto é frequentada por 30.000 nos horários de pico com uma projeção de crescimento para 70.000 pessoas até 2021. O MassMotion foi utilizado primeiramente para mapear o fluxo e comportamento das pessoas nas condições atuais. Com esses dados, foi possível gerar um relatório com possíveis pontos de melhoria para que a estação se adapte para o futuro. Essas medidas foram passadas para a prefeitura de Toronto.



Figura 3 - Simulação gerada pelo MassMotion [2]

O MASA Life [3] é um middleware que permite a criação de uma árvore de tomada de decisões para agentes autônomos incorporando comportamentos, pathfinding e movimentos. Essas árvores podem ser aproveitadas em aplicações dedicadas ou integradas a engine de jogos ou softwares de simulação distintos.

A Petrobras adotou o Simulex [4] como solução corporativa para evacuação de seus prédios. Produto proprietário, com boa adesão do mercado, é capaz de importar plantas (CAD) de vários andares para realizar uma simulação integrada. Pode-se ver a cena por diversos ângulos diferentes e configurar o perfil dos agentes por idade, sexo, altura, largura, velocidade e tempo de resposta ao alarme. Dependendo da configuração do usuário, o sistema não rodará em tempo real. Ele também é capaz de gerar um vídeo com a simulação completa e arquivos textos com o fluxo dos agentes.

A versão instalada atualmente na Petrobras não realiza simulações em 3D e não reproduz o fluxo das escadas. Por isso foi necessário o desenvolvimento de uma solução interna como complemento. Com isso, além de depender de dois softwares para realizar uma simulação completa, a mesma não é reproduzida em 3D e nem em ambientes multiescalas como refinarias ou plataformas.

2.2. Pesquisas acadêmicas

Existem diversos focos nas pesquisas direcionadas a simulação de multidões (crowd simulation). Otimizações para simulações em tempo real utilizando occlusion culling, reprodução de um fluxo contínuo de pedestres e modelagem de

comportamentos. Como essa dissertação se propõe a simular um evento de emergência, é necessária uma mescla de todas essas pesquisas.

Em [5], Tecchia, Loscos e Chrysanthou chamam atenção de que, apesar de simulação de multidão estar em destaque na comunidade, existem poucas aplicações que permitem uma manipulação em tempo real da cena. É destacada também a importância e dificuldade da representação do comportamento dos agentes.

Para otimização da detecção de colisão com o cenário e quais áreas do mesmo podem ser acessadas, o ambiente 3D é representado por um mapa de altura 2D em grid, evitando o uso de estruturas de dados mais custosas como bounding boxes, BSP trees e Octrees. Essa estrutura é totalmente carregada em memória e, através de uma variável configurável, os agentes saberão a altura máxima que podem ultrapassar.

Essa mesma estrutura é utilizada mais três vezes, sendo a segunda responsável pela colisão entre agentes. Antes de se movimentar, os agentes verificarão se a célula de destino já está ocupada. As terceiras e quartas camadas são responsáveis pelos comportamentos e ações. Cada região do grid tem um tipo de movimento que o agente deverá tomar variando entre parar, virar à esquerda, esperar um sinal e calcular uma nova rota. O foco do artigo está na representação de um fluxo contínuo de pedestres.

A dificuldade de representar muitos polígonos em tempo real é mencionada e toda a renderização é baseada em imagens pré-renderizadas. De acordo com os autores, técnicas como LOD e occlusion culling (processamento somente em CPU) não são muito eficazes em cenas com milhares de agentes autônomos. Atualmente, o uso de novas tecnologias, como processamento em GPU, provam ser possível esses tipos de técnicas.

Para melhorar a representação de uma multidão muito densa, Pelechano e Badler [6] propõem um modelo HiDAC (High Density Autonomous Crowds) onde os agentes se adaptam a ambientes dinâmicos e possuem diferentes tipos de comportamentos em uma mesma cena.

Ele pode simular um cenário de tranquilidade como o fim de uma sessão de cinema até situações de perigo extremo (evacuação de incêndio). Permite também reproduzir propagação de pânico, ultrapassagens, impaciência (um agente pode recalculá-la outra rota para fugir de uma fila), empurrões e agressividade.

O modelo permite a execução em tempo real com centenas de agentes ao mesmo tempo, porém a visualização gráfica não é muito real e detalhes técnicos do desenvolvimento não foram apresentados no artigo [6].

Para representar o comportamento de uma multidão numa evacuação, Zaharia et al propõem em [7] uma extensão do algoritmo de Boids [8]. Utilizando o conceito de agentes autônomos, os autores tentam representar o comportamento humano na movimentação de um ponto a outro em um cenário pré-determinado.

A solução apresentada é em 2D e, ao realizar a simulação, fatores de separação, coesão e alinhamento podem ser alterados em tempo real.

Boids é um conceito desenvolvido por Craig Reynolds em 1986 [8] com o intuito de simular um grupo de animais, como um bando de pássaros ou um cardume de peixes. Toda a sua lógica é baseada em geometria computacional tridimensional e três comportamentos de locomoção simples: separação (mudar de direção para evitar aglomerações), alinhamento (se dirigir para a média das direções do bando) e coesão (manter um posicionamento mediano ao resto do bando) – Figura 4.

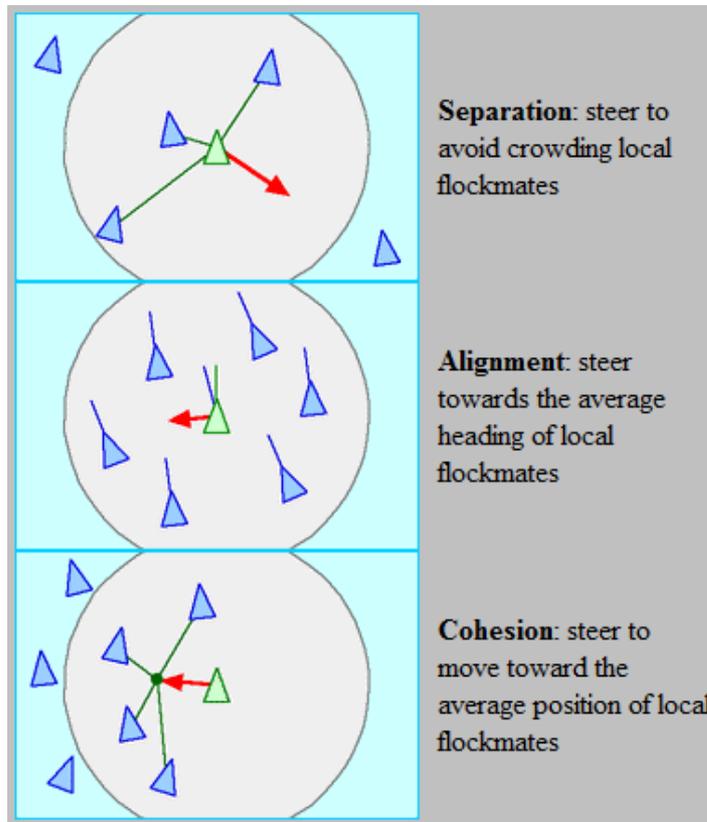


Figura 4 - Comportamentos de boids [8]

Esse comportamento de cada membro do rebanho é calculado de acordo com os seus vizinhos em uma determinada distância. É comum a inclusão de outras regras mais complexas, tais como detecção de obstáculos ou se dirigir a um ponto específico.

Esse conceito é muito utilizado na indústria de jogos e de cinema na representação de animais ou hordas de inimigos. Destaca-se o seu uso no filme *Batman O Retorno* (1992) na representação de um bando de morcegos e de um exército de pinguins. Essa técnica representou um grande salto na qualidade das animações cinematográficas, mostrando uma representação fiel à realidade, até então não presenciado pelo público.

Já Oguz et al apresentam em [9] uma nova forma de representação de uma multidão; o conceito de agentes é deixado de lado em prol de um enfoque em dinâmica contínua. De acordo com o autor, em uma emergência, as pessoas tendem ter um comportamento mais homogêneo, não há a necessidade de uma inteligência artificial individual. Com esse princípio, não há muita variação no comportamento dos avatares. O objetivo é puramente fugir do local acidentado.

Focado em ambientes externos, somente as pessoas localizadas a uma determinada distância da emergência irão fugir, as demais pessoas seguirão o trajeto pré-planejado, porém evitando a área afetada. A técnica de occlusion culling é utilizada na renderização.

A partir de estudos sociais e psicológicos, Luo et al [10] propõem um framework de comportamentos para multidões em situações normais e de emergência. Com módulos de características individuais e de grupo, os agentes podem ser configurados em perfis (funcionário, turista, civil), idade e gênero. Através de sua percepção do ambiente, um agente adotará um comportamento que está sujeito a uma ação. Eles realizam ações desde vagar, fugir, liderar, ajudar e correr.

No Brasil, é referência o trabalho feito na PUCRS pela professora Soraia Musse [11]. Com uma vasta gama de artigos, livros e publicações, várias perspectivas sobre simulação de multidões são analisadas.

Em um dos seus trabalhos, são simuladas evacuações de ambientes através da percepção de alarmes sonoros [12], onde cada agente é perceptível a uma determinada frequência, seja por idade ou distância do emissor sonoro. Desta forma, mostra-se a eficiência e problemas desse tipo de manifestação, já que

dependendo do ambiente a ser evacuado, podem existir diversos graus de percepções sensoriais distintas e com isso o alarme não chegaria a todos de forma uniforme. Geralmente uma evacuação deve ser orquestrada por sinais de formas diferentes (visual e sonoro) e por monitores ou brigadas de incêndio.

Em seu livro “Crowd Simulation” [13], Soraia explora os desafios de se realizar uma simulação de multidão. Entre os desafios, se destacam: a aparência realista e o comportamento dos agentes, estruturação de grupos, sua locomoção, detecção de obstáculos, e o mais complicado, tratar tudo isso alcançando resultados em tempo real. Métodos e técnicas são apresentados para ajudar a solucionar todas essas questões.

Em uma parceria com a Petrobras, foi desenvolvido o sistema PetroSim – Um Sistema para Simulação de Multidões em Situações de Emergência [14] no qual foi possível simular a evacuação da comunidade São José no Rio Grande do Norte. Com aproximadamente 120 casas e uma população de 450 pessoas foram definidos cenários de emergências e perfis de agentes (líderes comunitários, adultos e dependentes). Através de diversas configurações de perfis, foi possível definir a quantidade de feridos, mortos e salvos, sugerindo assim a melhor organização para a comunidade.

Em relação à área de exploração e produção de petróleo e gás natural, a ISO 15544 [15] estabelece a normatização de requisitos e diretrizes à resposta a emergências offshore. Em seu capítulo sobre fuga, refúgio, evacuação e resgate, informa a necessidade de estabelecer rotas de fugas de todos os lugares de trabalho, de locais de encontro para contagem de pessoas, procedimentos para evacuação total de forma ordenada e métodos de transporte das pessoas do local da emergência até outro local distante seguro.

Em outra parceria com o Botafogo de Futebol e Regatas, o Laboratório de Simulação de Humanos Virtuais da Faculdade de Informática da PUCRS (Facin) desenvolveu o CrowdSim [16]. O software é capaz de simular uma evacuação em locais com grande público, tais como, estádios, ginásios, prédios comerciais e residenciais, shows e festivais (Figuras 5 e 6).

Testado no Estádio Olímpico João Havelange, o Engenhão, no Rio de Janeiro, o software realizou a evacuação do local em seis minutos e trinta segundos em uma situação normal de final de partida com sua lotação máxima de 46.831 pessoas presentes e todas as saídas disponíveis. Já em outra situação, com

o caracol esquerdo do local interrompido, o tempo decorrido aumenta para sete minutos. O projeto, com os dados calculados, valida a segurança do estádio, já que nos resultados gerados, gargalos não foram encontrados mostrando assim a sua adequação a normas e padrões internacionais de segurança.

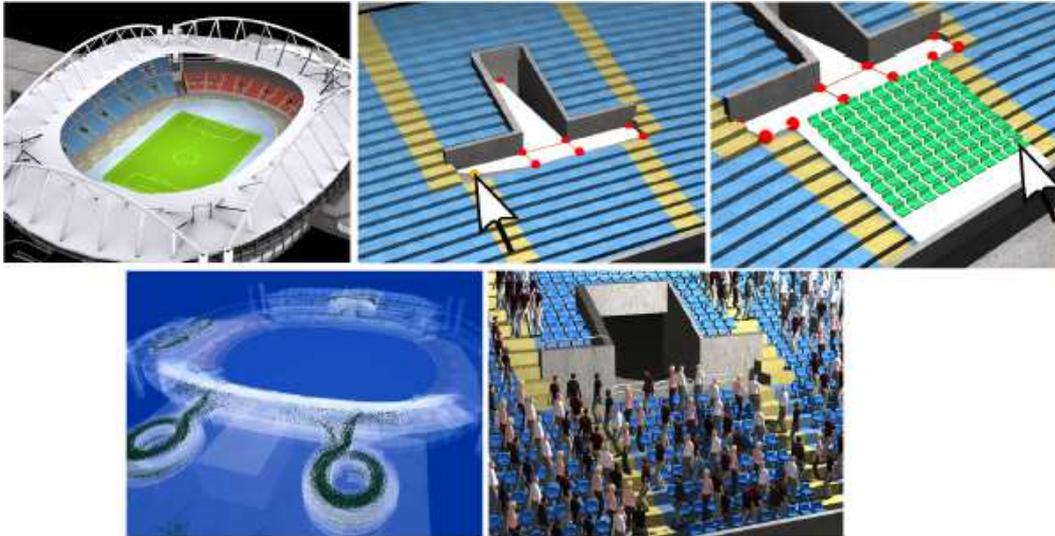


Figura 5 - CrowdSim executado no Engenhão [16]

Em um modelo em três dimensões, os usuários podem definir áreas de movimentação e circulação. Também são definidas as regiões onde os agentes devem se dirigir na evacuação. Baseado no conceito de fluxo de pedestres (vias) [17] [18], o software gera um arquivo para animação em softwares dedicados (3ds Max e Maya) e um mapa de densidade para análise mais detalhada de fluxos.



Figura 6 - Resultado do CrowdSim [16]

Em [17] é apresentado um modelo de forças na qual um aglomerado de pedestres é influenciado. Pedestres tendem a escolher a rota mais curta, geralmente uma reta mantendo uma velocidade constante e certa distância um do outro e de objetos na rua como paredes, obstáculos e meio-fio. Com esses modelos, mostram um comportamento padrão entre os pedestres, comportamento que tende a se repetir em outros cenários, tais como uma evacuação ou simulado de emergência.

Com base em tudo que foi apresentado, a intenção do presente trabalho é: apresentar um sistema corporativo [2] [4] de simulação de evacuação de emergência em cenários verdadeiros [14] [16] (especificamente em plataformas de petróleo [15]), comparando assim com o que acontece ou que pode acontecer na realidade; utilizar engines e plataformas do mercado sem necessidade de softwares de renderização dedicados [3] e algoritmos de otimização [5] [6] [7]; manipular a simulação em tempo de execução possibilitando observar o cenário de diversos ângulos diferentes e inserir comportamentos analisando o fluxo das pessoas na situação modelada [9] [10] [17] [18].

3

Método desenvolvido

Neste capítulo, é apresentada e justificada a plataforma de desenvolvimento escolhida. Também é detalhada toda a modelagem e configuração dos cenários e dos agentes e, por fim, o desenvolvimento do software proposto.

3.1.

Plataforma de desenvolvimento

Com um foco em criar uma aplicação, a ideia sempre foi aproveitar algo conhecido e aceito pelo mercado como apoio no desenvolvimento. Focar mais nas regras de negócio e na lógica para a situação a ser simulada do que em otimizações e renderização das imagens. Por isso, foi escolhido a engine de jogos Unity [19] como ecossistema para o trabalho desta dissertação.

O Unity é uma engine de jogos multiplataforma desenvolvida pela Unity Technologies. Principalmente utilizada em jogos mobile, consoles e desktops, ela é o SDK default de desenvolvimento para Wii U. Ela também começou a ser utilizada fora da indústria de entretenimento, como por exemplo, em aplicações de engenharia [20], arquitetura e construção, configuração de produtos e simulações médicas [21] e de segurança [22] [23].

A engine gráfica utiliza OpenGL e Direct3D e possui suporte nativo a textura, luzes, materiais, Occlusion Culling, LOD, sons, física e inteligência artificial. Com uma IDE integrada, toda a inteligência dos agentes é criada através de scripts em C# e Javascript.

Unity possui ferramentas nativas de construção de modelos geométricos e também integração com as principais aplicações de modelagem 3D. Importando e aproveitando, também, qualquer animação feita anteriormente utilizando uma máquina de estados.

Problemas enfrentados (malha de navegação, detecção de obstáculos, comportamento dos agentes, representação do coletivo, renderização em tempo real e modelos detalhados) pelos autores em [5], [6] e [7] poderão ser tratados

diretamente pela engine. Já que os cenários de emergência em uma plataforma de petróleo são mais controlados e restritos em relação ao número de agentes e de variáveis estruturais, os recursos presentes no Unity tendem a suprir as necessidades de processamento.

Portanto, por se tratar de um cenário de simulação de multidões mais discreto e pelas características do Unity, resultados em tempo real são possíveis e aguardados.

3.2. Modelagem

Para que uma simulação consiga refletir a realidade do ambiente, é de extrema importância a construção de modelos o mais próximo da realidade. Quanto maior a sua aderência, maiores chances dos resultados serem próximos do real.

Todos os modelos usados foram criados no 3ds Max e importados no Unity. Pode-se criar alguns objetos primitivos (caixas, esferas, cilindros, etc) para auxiliar o modelo em algum gargalo. O ideal é que não seja necessária nenhuma alteração e que caso seja realmente necessário, que as mudanças sejam feitas nas ferramentas de modelagem dedicadas.

Três modelos de plataformas foram construídos, cada um com um grau maior de detalhes. Isso afetará na qualidade dos resultados, podendo não refletir os tempos de uma evacuação em casos reais. Porém, computacionalmente, modelos mais simples tendem a ter uma performance de renderização melhor.

Dos três modelos utilizados, temos: uma plataforma semissubmersível com áreas internas e externas, com texturas e materias, uma plataforma FPSO em formato de navio somente com ambientes externos e texturas básicas (Figura 7), e uma plataforma FPSO redonda somente com ambientes externos e sem texturas. Todos os modelos são representações de plataformas reais e atuantes no mercado e com características diferentes (número de baleeiras e de pessoas embarcadas).

É importante alcançar esse ponto de equilíbrio entre o realismo do modelo e uma simplicidade para que o software consiga rodar numa taxa admissível de frames por segundo.

Com uma boa quantidade de modelos diferentes, saberemos se o algoritmo proposto é adequado para esses diversos cenários de simulação de evacuação de emergência.



Figura 7 - Plataforma do tipo FPSO em forma de navio

Para que os agentes consigam caminhar no modelo importado, o Unity permite a criação de uma malha de navegação (Navigation Meshes – NavMesh) – Figura 8. A malha criada nesse processo é o que permite que o algoritmo de pathfinding possa ser rodado para cada um dos agentes.

No Unity, o processo de criar essa malha se chama Bake. Para que você consiga executá-lo, deve-se informar no modelo quais planos ou figuras (chão e escadas) devem entrar no cálculo da navegação.

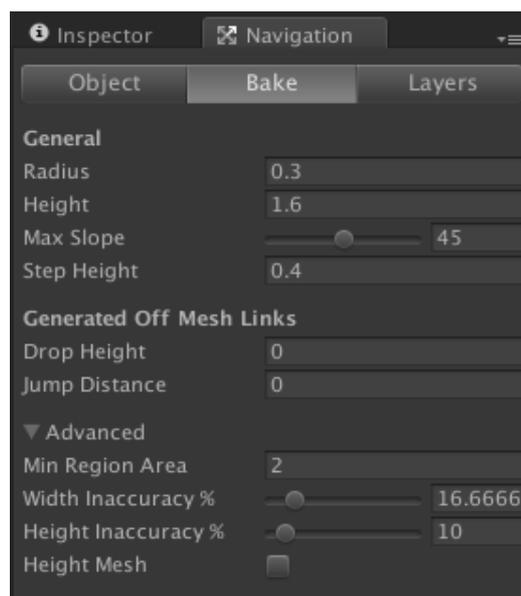


Figura 8 - Configuração da malha de navegação

Após informar os objetos, alguns outros fatores deverão ser definidos, possibilitando o cálculo de onde no modelo os agentes podem caminhar. O raio (Radius) determina o quão perto um agente pode chegar de uma parede e, conseqüentemente a largura mínima de um corredor. A altura (Height) define a altura mínima do teto, impossibilitando que um agente possa passar por um obstáculo sem se abaixar. A inclinação máxima (Max Slope) diferencia em graus uma parede de uma rampa. A altura do degrau (Step Height) define a altura máxima de um degrau, qualquer degrau maior do que esse será interpretado como um obstáculo pelo agente e a área não será alcançada. O salto e o pulo (Drop Height e Jump Distance) não são permitidos nos cenários simulados.

Nas Figuras 9 e 10, o NavMesh é representado pelas áreas em azul.

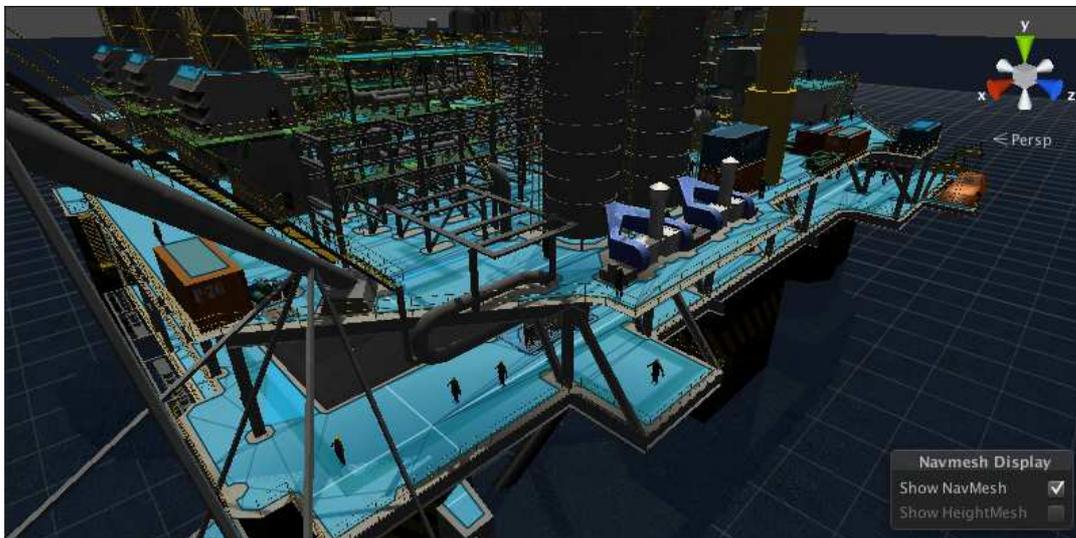


Figura 9 - Malha de navegação de uma plataforma semissubmersível



Figura 10 - Malha de navegação de um FPSO

Com o NavMesh finalizado, os agentes recebem um componente capaz de executar o pathfinding e sua movimentação. Esse componente é responsável por pegar as informações contidas na malha e calcular o caminho ótimo. Possuem diversas propriedades que definirão o seu comportamento com o ambiente e com os outros agentes.

Entre as principais propriedades estão: raio (Radius), que determina uma distância em volta do agente na qual nenhum obstáculo pode passar; velocidade (speed) e aceleração (Acceleration) máxima dos agentes em unidades por segundo; velocidade angular (Angular Speed) máxima dos agentes em graus por segundo; e distância de parada (Stopping Distance) estabelece a distância do alvo em que o agente pode parar – Figura 11.



Figura 11 - Configuração dos agentes

Só será usado um modelo de agente e suas características serão iguais para todos. Todos os parâmetros descritos acima serão idênticos. Como o cenário a ser simulado é bem restrito e específico, o perfil dos agentes não muda muito, tendendo para uma população masculina de estaturas e medidas médias.

Essas características é o que tem de mais importante nos agentes, já que os modelos 3D são interpretados como cilindros pela aplicação, não importando o movimento ou localização do agente durante a movimentação. A sua qualidade gráfica só serve para criar um ar de realidade na simulação, não afetando o resultado final (Figura 12).



Figura 12 - Agente renderizado

Além dessas características, os agentes possuem animações nos estados parado, andando e correndo.

3.3. Desenvolvimento

Apesar do Unity fornecer muitos recursos nativos, a aplicação ainda necessita de scripts para funcionar conforme o planejado. Durante o desenvolvimento, eles foram separados em três grupos: câmeras, controle e agentes.

3.3.1. Câmeras

Para que haja uma avaliação da simulação e do que está acontecendo em tempo real, as câmeras são de extrema importância. Elas permitem a visualização da cena como um todo, dos caminhos, tomados dos gargalos e pontos de melhoria nas estruturas. Portanto, foram criados três tipos de câmeras para tal.

Uma câmera em primeira pessoa, onde três agentes escolhidos aleatoriamente em lugares diferentes do cenário possuem uma câmera emulando sua visão durante todo o projeto permitindo uma análise no ponto de vista do indivíduo (Figura 13).

Uma visão de vigilância, dividindo a cena em quatro, possibilitando uma análise integrada, como por exemplo, de todas as baleeiras das plataformas ao mesmo tempo (Figura 14). Essa visão exige um grande poder de processamento, pois no fundo está renderizando quatro imagens distintas ao mesmo tempo, dividindo a tela do usuário em partes iguais. Quanto maior o detalhe da cena capturada por cada câmera, pior será a performance.



Figura 13 - Visão em primeira pessoa

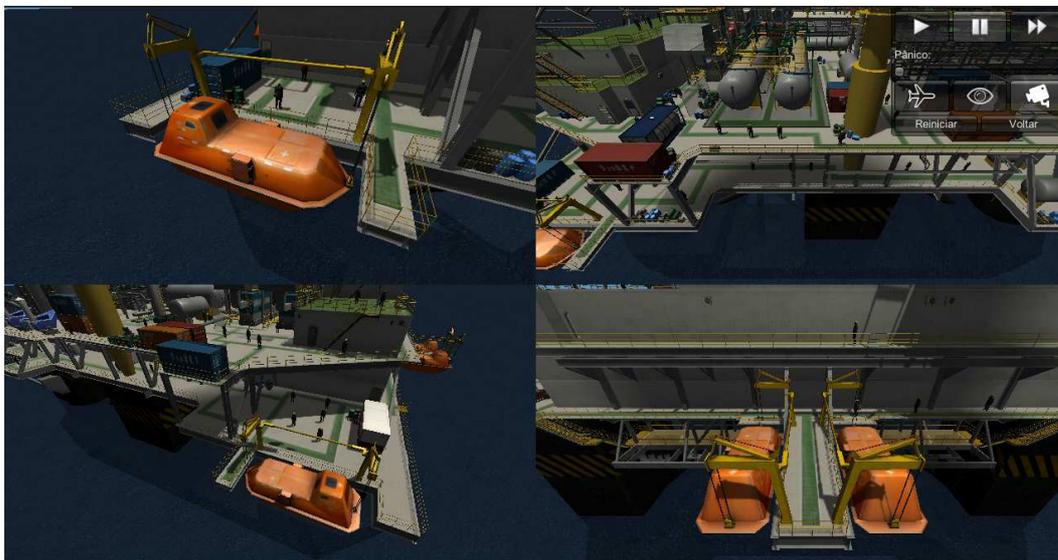


Figura 14 - Visão de vigilância com 4 câmeras

A principal câmera é do tipo “voo” (fly), como o nome sugere, com ela é possível voar todo o cenário em todas as direções e rotações e realizar zoom, verificando cada detalhe da evacuação.

Para obter o resultado desejado, a câmera central é uma adaptação de um controlador de personagem padrão de jogos 3D. A velocidade e aceleração foram aumentados para que percorra distâncias mais rapidamente que os agentes. A propriedade de pulo foi bloqueada e foram inseridas movimentação no eixo Y (para cima e para baixo). Por fim, como os personagens ficam presos ao chão, a gravidade do controlador foi zerada permitindo assim o voo da câmera em qualquer coordenada.

3.3.2. Agentes

Antes de falar sobre a inteligência dos agentes, é importante saber como funciona um cenário de evacuação ou abandono de uma plataforma real ou simulado.

Em qualquer cenário de evacuação, o objetivo é levar todos os indivíduos presentes no recinto a um local seguro no menor tempo possível. Para que isso aconteça, todas as pessoas deverão tomar uma rota, previamente estabelecida, até as saídas de emergência.

Essa descrição, apesar de parecer um pouco genérica, resume o básico de todas as evacuações de emergência. Porém, alguns fatores e especificações irão ditar o modo e como isso ocorrerá.

Primeiramente, podemos separar os cenários em espaços públicos e privativos. A grande diferença está no conhecimento prévio das pessoas presentes no local a ser evacuado. Em lugares privativos, a maioria das pessoas está acostumada com o local, muitas vezes utilizando diariamente. Em lugares públicos, o conhecimento do local entre os indivíduos pode ser grande ou até nulo.

Em ambos, as rotas devem ser muito bem sinalizadas, para que todos saibam e aprendam o caminho até o local seguro. Além de sinalizações (visuais e sonoras), é extremamente importante a participação de brigadas de incêndio ou de monitores. É natural que o tempo de evacuação de lugares conhecidos seja menor do que em locais com pessoas sem conhecimento prévio.

Os aspectos físicos e suas proporções também afetam a evacuação e com isso o modo de desenvolvimento da simulação, principalmente no número de

peças impactadas. Quanto maior o número de pessoas, mais complexa é a simulação, piorando ainda mais se juntarmos milhares que não conhecem o local a ser evacuado. O que não é o caso dos cenários testados.

Cada local também exige um determinado tipo de câmera para visualização. Em ambientes externos é muito mais fácil ter uma visão do todo, enquanto em ambientes fechados, se torna muito difícil ter a imagem de todos os locais ao mesmo tempo. Deste modo, é necessária a execução da simulação diversas vezes para que uma análise eficaz seja feita. A situação piora quando se tem uma mistura desses dois tipos de cenários.

Os casos tratados nesta dissertação são ambientes privativos, onde todos os agentes conhecem as rotas de fuga, tornando a inteligência mais direta. Também, por se tratar de locais com capacidades físicas restritas, otimizações gráficas (occlusion culling, LOD, rasterização, etc.), apesar de ajudar na eficiência, não são tão necessárias. Porém, apesar das limitações de tamanho, plataformas possuem ambientes fechados e abertos.

Dois cenários de emergência serão simulados para cada plataforma, onde em cada um, os agentes terão que se deslocar para lugares diferentes. Um cenário na qual todos irão para o heliponto da plataforma, emulando um caso de vazamento de gás e outro na qual cada agente irá se deslocar para uma baleeira específica, emulando um caso de incêndio. Em ambos os casos, o caminho mais curto será utilizado.

Por se tratar de cenários muito específicos, a simulação é pouco impactada por fatores externos. A população de uma plataforma é muito restrita e todos estão muito familiarizados com o local. Além disso, é prática em empresas do ramo, a realização de treinamentos e simulados de emergência com frequência. As rotas de fuga, geralmente, são bem mapeadas e sinalizadas. Maiores detalhes sobre o processo de evacuação em plataformas de petróleo serão explicados no capítulo de resultados.

Também existe a diferença entre uma evacuação e abandono. Em uma evacuação, devem ser retiradas todas as pessoas não envolvidas no controle da emergência, enquanto em um abandono, todas as pessoas devem ser retiradas da plataforma. O tipo de acidente ou emergência definirá o tipo de retirada. No caso do sistema desenvolvido, todas as pessoas participarão da simulação caracterizando uma situação de abandono.

Mesmo em ambientes bem sinalizados, com diversas rotas de fuga, com pessoas bem treinadas e equipes de resgate, o imprevisto pode dificultar o trâmite normal de uma evacuação. O tipo e o local do acidente podem impactar as condições físicas do ambiente, bloqueando uma passagem ou isolando uma quantidade de pessoas. Por isso, há a necessidade de um planejamento com inúmeros fatores, pensando em contingência de rotas e meios alternativos. Mesmo com todo planejamento, é muito difícil pensar em todas variáveis, mas um bom planejamento reduz bastante as chances de lidar com algo impensável.

No entanto, são as pessoas e seus sentimentos os maiores problemas e fatores a se lidar em emergências. Cada indivíduo é único e se comporta de modo diferente em situações extremas. Mesmo ela acostumada com o cenário e tendo participado de diversas simulações, é impossível prever como todas irão se comportar realmente numa situação de caos.

Simulados de emergência tendem a representar uma fuga ordenada, onde todos os agentes tendem a manter a mesma velocidade, sem correr, respeitando filas, obstáculos e outras pessoas. O ideal é que esse comportamento se repita mesmo em casos reais, possibilitando a fuga de todos sem problemas.

Porém é comum haver cenários com pânico, agressividade e força. Cada ser humano se comporta de um jeito na face de um perigo, prevalecendo o individual ao invés do coletivo. O seu comportamento no momento tende a ser intensificado com a sua percepção do perigo eminente. Pessoas em contato com o problema (fogo ou fumaça) terão reações mais impactantes.

Em prédios, onde as pessoas estão distribuídas em diversos andares, é comum essa divisão de comportamento. As pessoas nos andares onde a emergência está ocorrendo e em andares adjacentes devem ser evacuadas primeiro. Muitas vezes pessoas em andares distantes acham que estão participando de um simulado, pois não têm a percepção do perigo. Ficam sabendo da real situação quando chegam no local seguro.

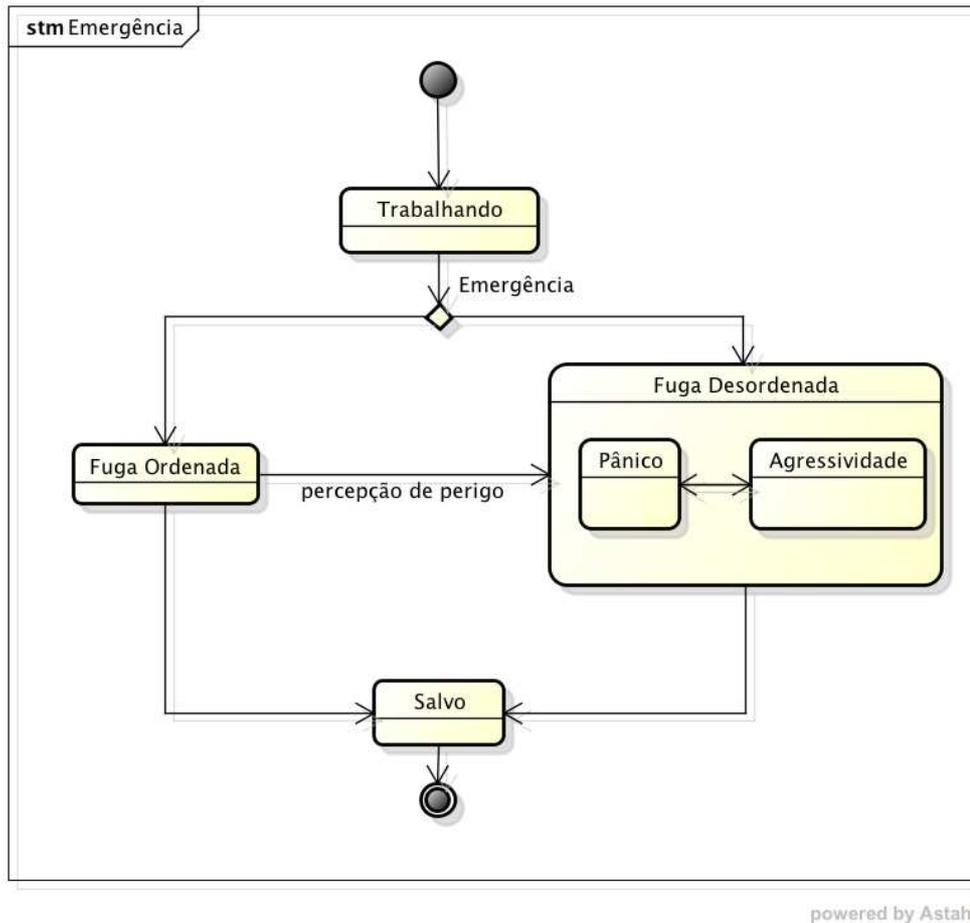
A Figura 15 apresenta os estados dos agentes durante toda a simulação de emergência.

Todos os agentes iniciam sua jornada como um dia normal, em seu devido local de trabalho executando suas devidas atividades. Ao iniciar uma emergência, eles podem realizar uma fuga ordenada ou desordenada. Uma fuga ordenada é realizada com calma, parecida com os simulados realizados com pessoas reais. Já

em uma fuga desordenada, os agentes tendem a demonstrar desespero, passando por sentimentos distintos, como pânico e agressividade.

A forma com que os agentes se comportarão no início da emergência é subjetiva, podendo variar de uma fuga ordenada ou desordenada de acordo com a sua percepção do perigo ou com sua capacidade de lidar com cenários extremos.

Ao chegarem ao local seguro pré-estabelecido, o agente estará a salvo alcançando assim o fim da simulação.



powered by Astah

Figura 15 - Estados em uma emergência

Para simular um grau de pânico, a velocidade dos agentes pode ser alterada no decorrer da simulação. Eles correrão numa velocidade máxima controlada, respeitando os outros indivíduos e obstáculos.

Toda a inteligência e comportamento dos agentes é feita através de scripts. Todos possuem o mesmo cérebro que é executado a cada frame, ou seja, em uma cena com 200 agentes rodando a 30 frames por segundo, os scripts serão chamados num total de 6000 vezes num segundo. Isso pode ser um problema, pois

essa inteligência concorre em concomitadamente com a renderização das imagens. O ideal é que qualquer cálculo maior seja feito numa fase de pré-processamento.

As principais funções (nativas do Unity) executadas em cada agente são: Awake, Start e Update.

A função Awake é sempre executada quando um objeto é inicializado na cena. Serão carregados componentes nativos do Unity responsáveis pelas animações, configurações do agente e pela malha de navegação que será usada para calcular a rota.

Sintaxe
<pre>function Awake() { animator = GetComponent.<Animator>(); agent = GetComponent.<NavMeshAgent>(); }</pre>

Tabela 1 - Função Awake

A função Start é chamada antes da execução do primeiro frame da aplicação. Será configurado a sua velocidade inicial e autoriza a transformação de sua orientação.

Sintaxe
<pre>function Start() { speed = GameObject.FindWithTag("Menu"). GetComponent(PainelControlePlataforma).GetSpeed(); agent.updateRotation = true; }</pre>

Tabela 2 - Função Start

A função Update é chamada uma vez por frame. Ela irá controlar a movimentação dos agentes através da variação de velocidade e verificará se o mesmo chegou no local seguro. Chegando a uma distância mínima deste local, o agente irá parar e mudará o seu estado para “salvo”.

Sintaxe

```
function Update() {
    var paused = GameObject.FindWithTag("Menu")
    .GetComponent(PainelControlePlataforma).GetStatus();
    if (paused == false) {
        if (safe)
            return;
        if (Vector3.Distance(transform.position,
goal.transform.position) <= 5.5) {
            animator.SetFloat("Speed", 0);
            agent.Stop();
            safe = true;
        }
        else {
            agent.Resume();
            speed = GameObject.FindWithTag("Menu")
            .GetComponent(PainelControlePlataforma).GetSpeed();
            animator.SetFloat("Speed", speed);
            agent.speed = speed;
        }
    }
    else {
        animator.SetFloat("Speed", 0);
        agent.Stop();
    }
}
```

Tabela 3 - Função Update

O conjunto dessas funções possibilita o comportamento e ação dos agentes durante a simulação.

3.3.3. Controle

O script de controle serve como maestro de toda a simulação, se comunicando com as câmeras, interface gráfica e com todos os agentes da cena.

Antes de iniciar qualquer processamento de pathfinding, o usuário deverá informar qual tipo de simulação será executada, neste caso, ou um cenário de incêndio ou de vazamento de gás.

A função mostrada na Tabela 4 é chamada para todos os agentes presentes na simulação, ela define qual o destino deles na simulação de acordo com a seleção do usuário. Após a definição do destino, cada agente irá calcular a sua rota individualmente. Este é o passo mais importante e custoso.

Sintaxe
<pre>function setGoal () { var tipo = GameObject.FindWithTag("Menu") .GetComponent(PainelControlePlataforma).GetTipo(); if (tipo == 0) { goal = GameObject.FindWithTag("GoalGas"); } else { goal = baleeira; } agent.SetDestination(goal.transform.position); safe = false; }</pre>

Tabela 4 - Função que define o local seguro

Após a seleção do cenário, o sistema só estará pronto a ser executado quando todos os agentes tiverem calculado a sua rota. Esse tempo não é preciso, porém, quanto maior o número de agentes na cena, mais longo será o tempo de processamento. A simulação não é considerada válida para início sem essa fase, pois nem todos os agentes da cena saberiam o que fazer.

O caminho calculado pelo Unity é um conjunto de pontos (waypoints) conectados por linhas retas, resultando no caminho mais curto do ponto inicial ao final. Por isso, os agentes ao invés de seguirem à risca esse trajeto, usarão uma locomoção mais real variando a velocidade, aceleração e velocidade angular, de acordo com as configurações realizadas no tópico anterior. Deste modo, tendem a manter uma locomoção menos robótica.

Por causa dessa locomoção mais natural, os agentes nunca irão parar exatamente no mesmo local marcado como ponto seguro. Isso acontece principalmente pela desaceleração e pela velocidade angular. Eles irão ultrapassar o ponto e recalculer uma nova rota até o mesmo.

Para que isso não ocorra eternamente, uma distância mínima de parada é estabelecida. Todos os agentes localizados a essa distância do alvo são considerados em uma local seguro e sua movimentação será parada.

A simulação será concluída e o cronômetro parará quando todos os agentes alcançarem esse estado.

A função abaixo irá percorrer todos os objetos da cena classificados como “Agente”. Caso exista algum que ainda não possui sua rota calculada (hasPath igual a falso), o sistema entende que ainda está em pré-processamento não liberando a simulação para execução.

Portanto isso causa um atraso no início da simulação, variando o tempo de pré-processamento de acordo com o modelo 3d utilizado e com o número de agentes a serem simulados. Passado esse período, a simulação será executada em tempo real.

Sintaxe
<pre>function loadScene () { var obj : GameObject[]; obj = GameObject.FindGameObjectsWithTag("Agente"); for (var o : GameObject in obj) { var agent : NavMeshAgent; agent = o.GetComponent.<NavMeshAgent>(); if (agent.hasPath == false) { loading = true; return; } } loading = false; }</pre>

Tabela 5 - Função que verifica o pathfinding

O Unity permite a visualização de todos os caminhos dos agentes, na Figura 16, eles são apresentados em vermelho. É possível perceber a linearidade das mesmas, caracterizando a escolha do caminho mais curto. Por causa disso, pode-se prever a concretização de filas.



Figura 16 - Rotas de fuga (em vermelho) dos agentes

Havendo a liberação (término do cálculo de todas as rotas), o usuário tem o poder de alterar a visão das câmeras em tempo real, assim como pausar, acelerar e reiniciar a simulação. A velocidade de todos os agentes pode ser alterada até a quantidade máxima de 4 m/s ou 14 km/h.

Todos esses controles são acionados através da interface gráfica mostrada na Figura 17.



Figura 17 - Painel de controle da cena

Com a arquitetura, modelagem e programação prontos, deve-se povoar as cenas com uma quantidade de agentes conforme os cenários reais, executar todas

as variáveis, comparar com a realidade, com pesquisas e com soluções adotadas no mercado.

4 Resultados

Neste capítulo é descrito o cenário onde a simulação é executada. São apresentados os resultados da simulação assim como o seu desempenho computacional. Finalmente, são demonstrados os pontos positivos e negativos do software desenvolvido.

Todas as simulações foram executadas em um Macbook Pro (modelo 2009) rodando OS X Yosemite e Unity 4.3 com um processador 2,26 GHz Intel Core 2 Duo, 4 GB 1067 MHz DDR3 de memória RAM e placa gráfica NVIDIA GeForce 9400M 256 MB.

4.1. Cénario simulado

A simulação foi realizada em modelos de 3 plataformas de petróleo distintas. Duas FPSO (Unidades Flutuantes de Produção, Armazenamento e Descarga) e uma plataforma semissubmersível – Figura 18. Cada uma com as características apresentadas na Tabela 6:

Plataforma	Quantidade de Pessoas	Baleeiras
FPSO Navio	200	4
FPSO Redonda	80	2
Semissubmersível	160	4

Tabela 6 - Configuração dos cenários simulados

Os três tipos diferentes de plataformas possibilitarão uma melhor análise do algoritmo desenvolvido, avaliando a sua adequação a modelos e quantidades de pessoas distintas. Cada plataforma possui uma particularidade geométrica e arquitetônica, exigindo esforço e movimentações diferenciadas dos agentes envolvidos na simulação.

A FPSO Navio se assemelha à um navio cargueiro, ou seja, é uma plataforma longa na qual os trabalhadores tendem a percorrer longas distâncias.

Seu modelo 3d tem qualidade mediana de detalhes, possui texturas básicas e as estruturas de produção presentes ao longo da plataforma não são acessíveis. A maior falha é a ausência dos ambientes internos (dormitórios, escritórios, etc). Para compensar essas faltas, todos os agentes foram espalhados por toda a extensão da plataforma obrigando os agentes a se locomoverem uma grande distância.

A FPSO Redonda é uma plataforma inovadora por ser em formato cilíndrico, sendo a menor das três avaliadas com poucas unidades semelhantes no mercado. O modelo 3d possuiu quase nenhuma textura e a localização das baleeiras é diferente em relação ao modelo original. Possui a mesma grande falha do modelo da FPSO Navio, ou seja, não possuiu os ambientes internos.



Figura 18 - Três modelos de plataformas utilizados

A Semissubmersível é o modelo de plataforma mais conhecido e recordado quando se pensa em plataformas de petróleo. Também é o modelo mais próximo à realidade. Possui boas texturas, ambientes internos e externos bem modelados e diversos níveis, estruturas e andares acessíveis aos agentes. Por causa dessa qualidade, os seus resultados tendem a ser os mais representativos, tanto para validação de tempo quanto de performance.

Essa variedade estrutural e gráfica dos modelos permitirá uma análise mais profunda do sistema desenvolvido, observando sua adequação e flexibilidade permitindo sua utilização em outros cenários.

Dois cenários de simulação estão disponíveis: abandono em caso de incêndio ou em caso de vazamento de gás.

Em caso de incêndio, cada agente deverá se deslocar para uma baleeira previamente configurada no momento do desenvolvimento da aplicação. Os agentes não se encaminharão para a baleeira mais próxima, pois no momento do embarque, cada trabalhador deve receber qual embarcação se dirigir em caso de emergência. Esse cenário é interessante, pois tende a ter conflitos de rotas, como por exemplo, alguns agentes usando uma mesma escada para subir ou para descer ou engarrafamentos em corredores estreitos.

Em um cenário de vazamento de gás, todos os agentes se encaminharão para o único heliponto da plataforma percorrendo o caminho mais curto do seu local de trabalho até o mesmo.

Inicialmente, todos os agentes estarão espalhados na plataforma em seus locais de trabalho retratando um dia normal de operação. Ao iniciar a simulação, todos eles se deslocarão andando, ao mesmo tempo, para o local de acordo com o cenário previamente escolhido.

Os dois cenários de evacuação também serão simulados nas três plataformas com todos os agentes correndo, passando uma ideia de pânico e de fuga desordenada. Apesar de todos terem a mesma velocidade, ela será alterada várias vezes durante o processo, mantendo um nível maior de realidade.

O abandono total não será representado, pois a descida das baleeiras ou resgate dos agentes não será implementado. Deste modo, o fim da simulação será representado com os agentes parados no local seguro aguardando as próximas instruções.

O cronômetro (tempo total da evacuação) se inicia no primeiro clique no “Play” e para automaticamente quando todos os agentes estiverem próximos ao destino definido em cada cenário.

4.2. Simulação contra a realidade

Com as cenas configuradas com o número de agentes adequados e com as rotas definidas, doze simulações foram executadas alcançando assim uma média de tempos, evitando alguma discrepância e falha eventual.

Os tempos médios apresentados na Tabela 7 foram alcançados nas três plataformas com todos os agentes andando na velocidade mínima configurada durante o tempo total da simulação.

Plataforma	Vazamento de Gás	Incêndio
FPSO Navio	6:13	5:33
FPSO Redonda	4:25	4:43
Semissubmersível	4:16	4:11

Tabela 7 - Tempo médio de evacuação (ordenada), em minutos

Em média, os tempos registrados em simulados com pessoas reais em plataformas ou sondas ficam entre 10 e 15 minutos. No caso dos modelos utilizados, as médias de tempo reais são: 14:30 para FPSO Navio, 10:00 para FPSO Redonda e 12:30 para a Semissubmersível.

Essa grande discrepância nos tempos é explicada devido alguns fatores. Existe um intervalo entre detecção do problema até o toque do alarme. Esse tempo deve ser curto, de no máximo um minuto. A percepção do alarme pelas pessoas e seu poder de reação também aumenta o tempo em um ou dois minutos em média.

No sistema desenvolvido, o alarme não é representado. Os agentes não possuem nenhum atraso, ou seja, iniciam a evacuação no exato momento que a simulação começa (botão Play). Portanto, só por esses detalhes, os tempos totais tendem a aumentar entre dois a três minutos.

Antes de se dirigirem ao ponto seguro para evacuação, todos deverão vestir qualquer equipamento de emergência ou segurança necessário (coletes salvavidas) e se dirigir ao Ponto de Reunião da plataforma. Lá será feita a contagem do efetivo verificando qualquer pessoa que falta. Só após essa etapa que todos deverão seguir para as baleeiras ou heliponto. Essa contagem é fundamental e rápida não excedendo cinco minutos. Ela garante a evacuação de todo o efetivo da plataforma, possibilitando atender alguma pessoa que tenha ficado para trás.

Todas essas etapas possuem tempos padrões, o sistema desenvolvido é essencialmente para calcular o tempo de locomoção na plataforma, ou seja, o tempo entre essas etapas. Esses tempos mudam bastante para cada tipo de plataforma, principalmente devido ao número de pessoas envolvidas e ao tipo de trajeto a ser tomado. Por se tratarem de estruturas diferentes, cada uma exige um esforço de locomoção diferenciado, como por exemplo, de acordo com os tipos de estruturas, número de andares e de escadas.

No caso da FPSO Redonda, somando os tempos padrões de acionamento do alarme, percepção e contagem das pessoas (1 + 2 + 5 minutos) com o tempo de locomoção alcançado pelo sistema (6:13), obtemos o tempo total de 14:13, ou seja, compatível com os tempos médios de simulados. O mesmo cálculo vale para os outros modelos de plataformas.

As médias apresentadas não englobam o abandono, ou seja, o tempo necessário para que as baleeiras sejam lançadas ao mar. Geralmente esse tempo fica em torno de 10 a 15 minutos.

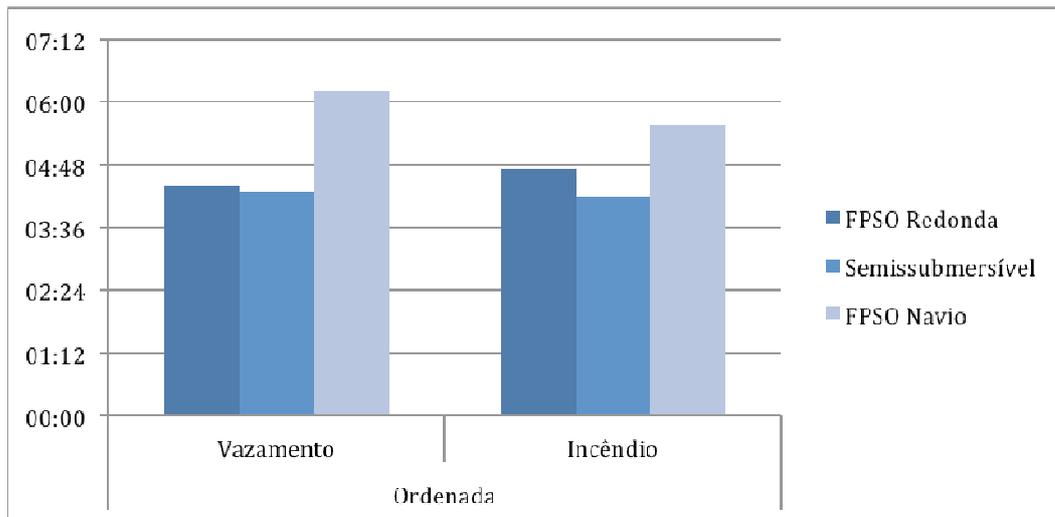


Figura 19 - Tempo médio de evacuação

Em casos reais, outros tipos de emergência também são planejados (explosões, acidentes com helicópteros, homem ao mar, invasões, etc), tendo cada hipótese um plano de evacuação diferente. Por se tratarem de casos e situações tão distintas, os tempos tendem a variar bastante. Porém é boa prática que a evacuação não estoure os tempos apresentados. Os chefes das plataformas devem decidir se cabe ou não uma evacuação para cada tipo de emergência. Nem todo incêndio deverá levar ao abandono. Essa decisão pode ser tomada durante a contagem nos Pontos de Reunião.

Condições climáticas também causam transtornos podendo impactar uma evacuação em horas. Em casos de tempestade ou mal tempo, qualquer locomoção aérea é proibida.

Uma emergência também pode durar horas ou dias. Recentemente, em um caso de grande impacto e destaque da mídia, a plataforma Deepwater Horizon² afundou, em 2010, no Golfo do México dois dias após sofrer uma explosão. Das 126 pessoas a bordo, 11 faleceram.

Os tempos médios apresentados na Tabela 8 foram alcançados nas três plataformas com todos os agentes correndo em velocidades variadas, da mínima até a máxima – Figura 20. O tempo de corrida em cada velocidade é aleatório.

Plataforma	Vazamento de Gás	Incêndio
FPSO Navio	3:49	3:05
FPSO Redonda	2:05	2:16
Semissubmersível	2:58	2:17

Tabela 8 - Tempo médio de evacuação (desordenada), em minutos

Apesar da grande redução de tempo nas simulações com os agentes correndo, isso não quer dizer que esse deve ser o comportamento dos mesmos em uma situação de emergência. Corrida e aumentos esporádicos de velocidade não retratam realmente uma fuga desordenada. Além do mais, em uma real situação de emergência, a calma deve ser prezada ao máximo evitando qualquer tipo de caos.

Para retratar realmente um estado de fuga desordenada, outros tipos de comportamentos seriam necessários. Pessoas perdidas, em pânico ou atordoadas impactariam o todo, aumentando o tempo total de evacuação e no comportamento dos demais agentes.

Também, apesar do universo de plataformas ser mais restrito, é natural a diferença física dos trabalhadores, como por exemplo, mulheres, pessoas acima do peso, muito altas ou magras. Cada tipo tende a se comportar diferentemente, mudando principalmente o padrão de locomoção. Por se tratar de um espaço industrial, não é comum a presença de portadores de deficiência. Como, na simulação, os agentes são todos iguais, eles foram configurados para representar a média desse universo de trabalhadores.

² Fonte: http://en.wikipedia.org/wiki/Deepwater_Horizon

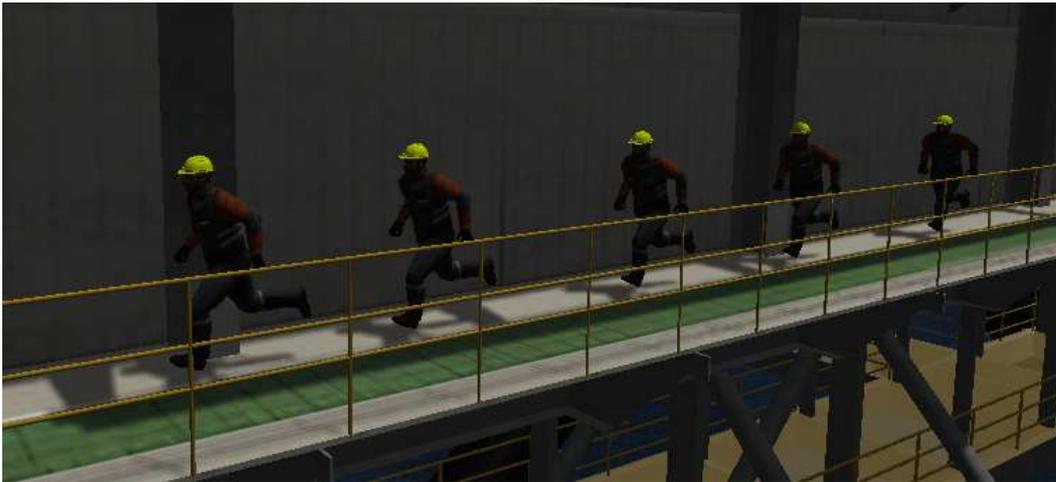


Figura 20 - Agentes correndo

Outro fator a se considerar, são as vestimentas das pessoas, dificultando sua locomoção e prováveis equipamentos que estejam sendo carregados ou manipulados no momento da emergência. Todos esses fatores impactam a evacuação alterando o seu tempo total.

Em ambas as simulações, comportamentos semelhantes ao de pedestres foram observados nos agentes, como retratado em [10], [17] e [18]. Grandes filas foram formadas resultando em aglomerados em portas, escadas e corredores – Figura 21. O comportamento homogêneo defendido em *Emergency Crowd Simulation for Outdoor Environments* [9] também foi alcançado, demonstrando ações coletivas e não individuais.

Essas representações foram possíveis principalmente devido às rotas calculadas no pathfinding, as animações representando uma movimentação mais humana, a detecção de obstáculos e interação com outros agentes. Ao calcular caminhos mais curtos, é natural que as rotas coincidam a partir de um ponto do cenário. Desta forma, filas serão formadas e representarão o comportamento humano.

Em uma evacuação de emergência, não é recomendada a alteração de rota no decorrer da fuga. Qualquer mudança no comportamento pode afetar o resultado final de uma evacuação. Isso somente deve acontecer em casos extremos, onde a rota previamente mapeada e conhecida esteja obstruída, não havendo alternativa.

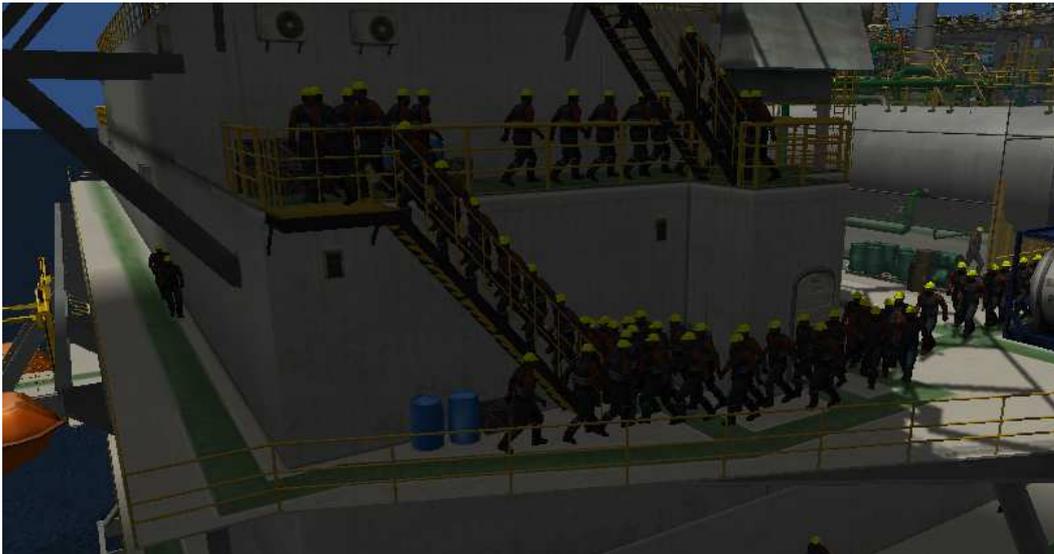


Figura 21 - Filas e aglomerados

Apesar dessas questões, os tempos alcançados foram os esperados, atingindo as expectativas e conseguindo retratar a realidade com certo grau de fidelidade. Incluindo uma margem de tempo para problemas e questões não desenvolvidas (tempo de reação, perfis de agentes e equipamentos), os tempos condizem com o mundo retratado.

4.3. Desempenho do software

Todos os dados apresentados a seguir foram registrados com o auxílio de uma ferramenta de apoio nativa do Unity capaz de informar as estatísticas de renderização das cenas simuladas em tempo real.

Os dados da Tabela 9 foram registrados nas três plataformas, durante fugas ordenadas e desordenadas, utilizando as câmeras de voo e em primeira pessoa em diversos momentos e lugares diferentes dos modelos. O tipo de fuga não causa impacto significativo na taxa de frames por segundos.

Plataforma	Média de Triângulos ³	Câmera Central	Visão Primeira Pessoa
FPSO Navio	3 milhões	24 fps	20 fps

³ Valores calculados pelo Unity sofrem influência de shaders, luzes e materiais. Não representa o número total real de triângulos dos modelos.

FPSO Redonda	1,4 milhões	30 fps	50 fps
Semissubmersível	4 milhões	9 fps	15 fps

Tabela 9 - Frames por segundo (FPS) por câmera

Os resultados alcançados permitiram a execução de todos os cenários em tempo real, principalmente nas FPSO Navio e Redonda. O desempenho alcançado na plataforma Semissubmersível foi fraco, porém ainda consegue-se analisar a simulação em tempo de execução.

Percebe-se uma grande relação entre o número de triângulos com a performance da aplicação. Por se tratar do modelo mais detalhado entre os utilizados, a qualidade do modelo da plataforma semissubmersível impactou bastante nos frames por segundo registrados. Reduzindo o número de triângulos e geometrias, texturas e iluminação, o resultado tende a melhorar, chegando ao nível das demais plataformas. A defasagem do hardware utilizado também possui grande influência nesses resultados, o uso de um processador e de uma placa de vídeo mais modernos aumentaria a concorrência na execução dos scripts e renderização.

As médias de frames por segundo na visão de vigilância não foram apresentadas, pois foi observada em testes uma grande queda de desempenho. Como a tela é dividida em quatro partes iguais, quatro imagens diferentes são renderizadas ao mesmo tempo. Portanto, a sua média de FPS tende a ser um quarto da média da câmera central. O cenário piora caso todas as câmeras estejam apontadas para partes da plataforma com um alto grau de detalhes (geometrias e texturas) e com grande presença de agentes.

Outro fator analisado bastante importante é o de pré-processamento, que é principalmente impactado pelo cálculo do pathfinding. Neste sistema, entende-se pré-processamento pelo tempo necessário para que as simulações estejam aptas para iniciar.

A Tabela 10 apresenta os tempos foram observados para os cenários previamente detalhados:

Plataforma	Vazamento de Gás	Incêndio
FPSO Navio	83,67	55,30
FPSO Redonda	68,42	30,81
Semissubmersível	222,31	178,11

Tabela 10 - Tempo médio de pré-processamento em segundos

Analisando os dados, percebe-se que o tempo total de cálculo do pathfinding em simulações de evacuação em caso de incêndio é menor que no cenário de vazamento de gás. Isso acontece, pois os caminhos até as baleeiras (Figura 22) tendem a ser mais simples sem zigue-zague. Com isso, o número de waypoints é menor, resultando num caminho mais reto. Enquanto em casos de vazamentos de gás, onde os agentes devem se direcionar até o heliponto, os caminhos tendem a possuir várias escadas, ou seja, uma rota mais custosa.



Figura 22 - Fim da simulação

No sistema desenvolvido, o tempo de pathfinding é calculado ao mesmo tempo em que a plataforma é renderizada para o usuário. Com isso, existe concorrência de processamento na renderização e na execução dos scripts dos agentes. Assim, quanto mais detalhado e complexo o modelo, menor a taxa de frames por segundo e maior o tempo de pré-processamento.

Portanto, os modelos e número de agentes estão diretamente relacionados ao cálculo total do pathfinding, ou seja, qualquer melhoria feita para aumentar o número de frames por segundo diminuirá o tempo de pré-processamento.

Outro ganho poderia ser alcançado se todas as rotas fossem calculadas sem nenhuma imagem sendo renderizada ao mesmo tempo, desta forma, todo esforço computacional estaria focado no pathfinding.

Assim sendo, um bom desempenho foi alcançado com bastante espaço para melhoria, tanto na questão de modelagem quanto na programação dos scripts de controle.

4.4. Benefícios, melhorias e gargalos

Com os resultados apresentados anteriormente, obteve-se sucesso tanto no negócio retratado quanto na tecnologia e arquitetura escolhida. O Unity permite representar uma simulação de evacuação de emergência com qualidade gráfica adequada em tempo real, apesar de não ter um grande número de agentes e se distanciar um pouco de soluções de multidões (crowd simulation) do mercado [1] [2] [3] [4].

Dependendo das regras de negócio a serem representadas e com os devidos ajustes e desenvolvimentos, resultados semelhantes e até melhores podem ser alcançados. Cada cenário e ambiente é melhor adaptado e representado por soluções tecnológicas diferentes [5].

Alguns gargalos e pontos de melhorias foram detectados para serem aplicados nas diversas etapas do processo de desenvolvimento. Os scripts default do Unity permitem um enorme ganho de produtividade, construindo cenas altamente detalhadas de forma rápida sem preocupação em algoritmos de renderização como rasterização e no cálculo de rotas. Recursos gráficos como Occlusion Culling e Level of Detail (LOD) estão disponíveis, porém não foram utilizados. Scripts especializados poderão trazer soluções mais eficientes.

Porém esses benefícios trazem complicações caso alguma extensão na sua lógica seja necessária. É permitido o desenvolvimento de pacotes para otimização ou alteração de algum comportamento. Por exemplo, ao utilizar o pathfinding nativo, utilizamos um algoritmo simples de busca do caminho mais curto. Em uma simulação de evacuação, essa lógica pode não representar as rotas que deveriam ser tomadas, podendo causar gargalos e amontoados de pessoas. Uma inteligência pode ser feita para que agentes evitem rotas muito congestionadas, recalculando um novo caminho e resultando numa melhor evacuação.

O grau de realidade a ser representando também é um fator importante de análise. Se a simulação for utilizada somente para análise, um modelo não precisa

ter qualidade gráfica próxima da realidade. O mais importante é que ele retrate a engenharia e arquitetura do local fielmente. Texturas e iluminações não são totalmente necessárias. Figuras geométricas simples com menos polígonos devem ser favorecidas também.

O nível de complexidade gráfica do modelo impacta diretamente no desempenho da aplicação e na taxa de frames por segundo, o que é crucial para execução da simulação em tempo real. Essa relação fica clara nos resultados alcançados nos três modelos de plataforma utilizados. Portanto, cabe definir as prioridades e foco para o caso a ser simulado, gráficos bonitos ou modelos representativos.

Para ilustrar um cenário real, os comportamentos e variedade de perfis de agente é essencial. Contudo, simular o comportamento humano é difícil e complexo, sobrando sempre uma brecha para o inesperado.

Ao configurar os agentes com medidas médias da população (altura, largura, velocidade e aceleração) consegue-se representar um universo grande sem muito trabalho de desenvolvimento. Ao representar somente um modelo, não é necessária a modelagem de vários agentes diferentes simplificando a simulação.

No entanto, a uniformidade de comportamentos e ações atrapalha na representação real da evacuação. A simulação tende a ficar constante, sem brecha para alguma situação conflitante ou inusitada.

O ideal seria o desenvolvimento de diversos agentes diferentes, representando toda a gama da população a ser simulada. O conflito deles durante a evacuação de emergência aproximaria a simulação da realidade retratada.

Além dos perfis, comportamentos também são essenciais. Os comportamentos desenvolvidos conseguem retratar um cenário de simulado (treinamento) fielmente, porém em casos de perigo as pessoas tendem a se comportar de modo diferente.

Apesar da calma ser essencial para o decorrer de uma boa evacuação, os seres humanos são imprevisíveis. Qualquer ação inesperada pode atrapalhar o conjunto, impactando os outros indivíduos presentes ao redor. Por isso, quanto mais comportamentos forem desenvolvidos (pânico, desmaio, brigas, desespero, etc.), mais próximo da realidade ficará. Mesmo assim, esses comportamentos não devem ser usados em excesso, o ideal seria uma frequência pequena e aleatória

dos mesmos. Resultados alcançados em [6] e [10] podem aproximar o trabalho desenvolvido à realidade.

O local onde a emergência ocorre é fundamental. Diversos tipos de emergência exigem estratégias de evacuação distintas. Prioridades e decisões são tomadas para que o tratamento às pessoas mais impactadas seja prioritário. Desta forma, cada grupo presente no local tende a ser evacuado em tempos diferentes, podendo afetar o tempo total.

Além de afetar a estratégia, o local da emergência também pode afetar alguma estrutura ou passagem. Essa alteração impacta na rota que os agentes tomarão. Geralmente as pessoas somente saberão desse impacto ao se depararem com a obstrução do caminho, portanto deverão tomar (recalcular) uma nova rota em tempo real (tempo de execução). Para isso é essencial a existência de rotas paralelas e alternativas.

É importante também sua evolução de forma que retrate as demais etapas da evacuação: acionamento do alarme, percepção sonora ou do perigo, coleta de equipamento e contagem do efetivo nos Pontos de Reunião. Cada etapa dessa traz outros desafios específicos, como o apresentado em [12], possibilitando trabalhos e soluções especializadas.

O realismo nas movimentações é retratado através das animações dos agentes. Quanto melhor animado, melhor será a adequação, da simulação à realidade. Existe uma carência atualmente relacionada às escadas, pois não foi animado o movimento de subir e descer um degrau. Portanto, a visualização das escadas ficará parecida com o movimento de subir e descer uma rampa. O sistema de pathfinding padrão do Unity também favorece que a movimentação fique desse jeito.

Com tudo que foi exposto, existe bastante espaço para melhorias em todas as etapas de desenvolvimento de software, porém o sistema desenvolvido consegue retratar logicamente e tecnicamente o cenário simulado.

5 Conclusão e trabalhos futuros

Simulação de multidões está em voga no cenário acadêmico e comercial. Uma de suas grandes aplicações está na representação de pedestres, evacuações e cenários de emergência, de modo que consiga prevenir algum possível problema ou falha.

Segurança cada dia que passa está mais presente na mente das pessoas e na visão estratégica de empresas e governos. Sistemas de informações podem e devem auxiliar qualquer área a testar e prever situações e gargalos que possam causar perigo a qualquer pessoa presente no local.

Juntando esses dois conceitos, essa dissertação apresentou uma proposta expansível, de fácil e rápido desenvolvimento, capaz de simular cenários de evacuação de emergência em plataformas de petróleo.

Doze simulações foram realizadas com três modelos de plataformas diferentes, cada um com uma especificação particular. Todos os resultados foram conforme o esperado, conseguindo retratar uma situação próxima a realidade com qualidade gráfica e de desempenho.

Porém, nem todas as etapas de uma evacuação foram desenvolvidas. Apesar de calcular principalmente as distâncias e os trajetos percorridos pelas pessoas, é interessante o desenvolvimento completo do negócio. Desta forma, haveria a representação fiel do passo a passo da emergência em uma plataforma de petróleo.

A complexidade dos cenários e os diversos tipos de emergência que podem ocorrer em uma plataforma de petróleo dificultam a representação da realidade. Diversas variáveis precisariam ser modeladas e desenvolvidas para uma total representação. Esse esforço de implementação tem que ser analisado e medido, pois certos tempos tendem a manter um padrão não impactando nos resultados finais.

A plataforma de desenvolvimento escolhida possibilita um novo foco no desenvolvimento de aplicações de simulação. O Unity foi escolhido por ser uma IDE (integrated development environment) barata e de fácil acesso, porém abre

portas para que outras plataformas de jogos mais robustas possam ser utilizadas. Engines mais poderosas, como a Unreal Engine⁴, são utilizadas na indústria de jogos com sucesso, representando cenas com milhares de agentes autônomos em tempo real.

Mesmo utilizando o Unity, diversos pontos de melhorias foram apresentados trazendo perspectivas positivas para a aplicação do sistema desenvolvido em trabalhos futuros.

Simulações de evacuações em cenários mais complexos, como, em prédios comerciais, estádios de futebol ou em casas de show aproximaria a solução proposta a um caso de crowd simulation. Representar um grande número de agentes ao mesmo tempo é um grande desafio. Simular casos mais complexos possibilitará enxergar a compatibilidade e aderência desse tipo de plataforma de desenvolvimento a este nicho de pesquisa.

A aplicação de comportamentos e tipos de emergência traria mais complexidade e realismo às evacuações. Desta forma, o sistema poderia ser aplicado em outros cenários, por exemplo, ataques terroristas ou manifestações civis. A movimentação dos agentes parecida com o fluxo de pedestres permite aproveitar o sistema em eventos públicos e do cotidiano, auxiliando a tomada de decisões.

Conclui-se que sistemas desse tipo podem retratar a realidade possibilitando análise de gargalos, padrões e comportamentos que podem acontecer numa situação real de emergência. Apesar do Unity exibir os caminhos dos personagens, ferramentas para análise de gargalos (gráficos de velocidade, mapas de calor e logs) são importantes para uma melhor avaliação do todo.

Também podem ser adaptados para retratar qualquer situação com presença de indivíduos autônomos em diversos tpos de situação. A partir de estudos, mudanças podem ser feitas trazendo benefícios, ganhos financeiros, adequação a padrões e normas e até, principalmente, salvando-se vidas.

Mesmo sendo muito difícil retratar a realidade, simulações de emergência podem evitar uma tragédia, porém eles sozinhos não fazem milagre. Qualquer ganho já é imensurável. Uma constante política de conscientização deve estar presente em qualquer ambiente, comercial ou não. Rotas devem ser bem

⁴ Fonte: <https://www.unrealengine.com/>

mapeadas e sinalizadas. Uma análise de impacto deve ser feita antes e após qualquer mudança arquitetural. Constante execução de treinamentos e simulados de emergência são essenciais para o preparativo das pessoas, assim como a constante equipe de apoio preparada para o combate. Tudo tem que estar preparado para o pior, mesmo que ele nunca aconteça.

6

Referências bibliográficas

- [1] Massive Software. **Massive (Multiple Agent Simulation System in Virtual Environment)**. Disponível em: <http://www.massivesoftware.com>
- [2] Oasys Software. **MASSMOTION**. Disponível em: <http://www.oasys-software.com>
- [3] MASA Group. **MASA life**. Disponível em: <http://www.masagroup.net>
- [4] Integrated Environmental Solutions Limited (IES). **Simulex**. Disponível em: <http://www.iesve.com/software/ve-pro/analysis-tools/egress/simulex>
- [5] TECCHIA, F., LOSCOS, C., CHRYSANTHOU, Y. 2002. **Visualizing Crowds in Real-Time**. In Computer Graphics Forum
- [6] PELECHANO, N., BADLER, N. 2006. **Improving the Realism of Agent Movement for High Density Crowd Simulation**. University of Pennsylvania, Center of Human Modeling and Simulation
- [7] ZAHARIA, M., LEON, F., PAL, C., PAGU, G. 2011. **Agent-based Simulation of Crowd Evacuation Behavior**. In 11th WSEAS International Conference on Automatic Control, Modeling and Simulation
- [8] REYNOLDS. C. W., 1987. **Flocks, Herds, and Schools: A Distributed Behavioral Model**. In Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.

- [9] OGUZ, O., AKAYDIN, A., YILMAZ, T., GUDUKBAY, U. 2009. **Emergency Crowd Simulation for Outdoor Environments**. In Computer & Graphics
- [10] LUO, L., ZHOU, S., CAI, W., LOW, M., TIAN, F., XIAO, X. WANG, Y., CHEN, D. 2008. **Agent-based Human Behavior Modeling for Crowd Simulation**. In Wiley InterScience
- [11] Musse, Soraia R., Publicações e trabalhos disponíveis em <http://www.inf.pucrs.br/~smusse/Journals.html>
- [12] CONY, C., DA SILVA, B., JUNG, C., MUSSE, S. SBGames 2006. **Simulação de Agentes com Capacidade Auditiva em Situações de Emergência**
- [13] MUSSE, S. R, THALMANN, D., 2007. **Crowd Simulation**. Springer-Verlag London Ltd
- [14] Musse, S. R. et al. 2005. **PetroSim - Um Sistema para Simulação de Multidões em Situações de Emergência**
- [15] ISO - International Organization of Standardization. **Petroleum and Natural Gas Industries - Offshore Production Installations - Requirements and Guidelines for Emergency Response**. ISO 15544:2000
- [16] MUSSE, S. R., CASSOL, V. J., RODRIGUES, R. A. et al. 2012. **CrowdSim: Uma ferramenta desenvolvida para Simulação de Multidões**. XI SBGames November 2nd - 4th, 2012
- [17] HELNBING, D., MOLNAR, P. 1998. **Self-Organization Phenomena in Pedestrian Crowds**. ArXiv Condensed Matter e-prints (June)
- [18] STILL, G. K. 2000. **Crowd Dynamics**. PhD thesis, University of Warwick, Coventry, UK

- [19] Unity Technologies. **Unity**. Disponível em: <http://unity3d.com>
- [20] F. Carvalho, A. Raposo; I. Santos, M. Galassi, 2014. Virtual Reality techniques for planning the Offshore Robotizing. 12th IEEE International Conference on Industrial Informatics (INDIN), p. 353-358.
- [21] P. Gamito, J. Oliveira et al. 2001. Traumatic Brain Injury memory training: a Virtual Reality online solution. International Journal on Disability and Human Development 10(4): 309-312.
- [22] S. Sharma, S. Jerripothula, S. Mackey, and O. Soumare, 2014. Immersive virtual reality environment of a subway evacuation on a cloud for disaster preparedness and response training. IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI), p. 1-6.
- [23] Silva, José Fernando, João Emílio Almeida, Rosaldo JF Rossetti, and A. Leca Coelho. "A serious game for EVAcuation training." In Serious Games and Applications for Health (SeGAH), 2013 IEEE 2nd International Conference on, pp. 1-6. IEEE, 2013