

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

### A 3D Simulation System for Emergency Evacuation in Offshore Platforms

Guilherme Bezerra Zampronio

DI / PUC-Rio  
Rio de Janeiro, Brasil  
zampronio@gmail.com

Alberto Barbosa Raposo

DI / PUC-Rio  
Rio de Janeiro, Brasil  
abraposo@tegraf.puc-rio.br

Marcelo Gattass

DI / PUC-Rio  
Rio de Janeiro, Brasil  
mgattass@tegraf.puc-rio.br

**Resumo**—Uma aplicação de simulação de evacuações, usando recursos computacionais, pode ajudar a prever situações, fluxos, conflitos e comportamentos que somente acontecerão numa 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. Este artigo 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.

*Simulação de Emergência; Multidão; Engine de Jogos; Petróleo e Gás; 3D*

**Abstract**—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 paper 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.

*Emergency Simulation; Crowd; Game Engines; Oil & Gas; 3D*

#### I. 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.

Em ambientes empresariais, as regras tendem 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 verificação de gargalos, possibilitando a criação de rotas alternativas atendendo o fluxo total de pessoas presentes no local em um tempo de evacuação adequado.

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 um grupo de pessoas é extremamente difícil e precisa de certos cuidados. Todo grupo de pessoas é formado 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 de multidões, em situações de emergência ou não [1]. 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.

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 privados. A grande diferença está no conhecimento prévio das pessoas presentes no local a ser evacuado. Em lugares privados, 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.

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 pessoas 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.

Os casos tratados neste trabalho são ambientes privativos, onde todos os agentes conhecem as rotas de fuga, tornando a inteligência mais direta, não necessitando de impulsos sonoros ou visuais.

O principal objetivo deste trabalho 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. Resultados em tempo real não são obrigatórios, porém bem-vindos.

Este artigo está organizado da seguinte forma. A Seção II discute os trabalhos relacionados, apresentando aplicações comerciais e corporativas e pesquisas acadêmicas. As Seções III e IV apresentam a arquitetura, e a implementação da aplicação, respectivamente. Os resultados e um comparativo com a realidade e com os trabalhos relacionados são descritos na seção V. Finalmente, a Seção VI apresenta as conclusões e aponta possibilidades de trabalhos futuros.

## II. TRABALHOS RELACIONADOS

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 [2] virou referência na indústria de entretenimento. Ele surgiu da necessidade de representar diversas batalhas entre centenas de milhares de soldados, problema até então não resolvido pela indústria cinematográfica.

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.

Outro software comercial de sucesso, porém focado em planejamento urbano é o MassMotion [3]. 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.

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.

Em [4], os autores 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 trabalho 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.

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

Testado no Estádio Olímpico João Havelange, o Engenho, 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 gargalos não foram encontrados, mostrando assim a sua adequação a normas e padrões internacionais de segurança. 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.

O principal objetivo do presente artigo é apresentar uma proposta de implementação de um sistema de simulação de evacuação de pessoas em uma situação de emergência em plataformas de petróleo com qualidade gráfica, com resultados próximos a realidade, utilizando como base, uma engine de jogos aceita pelo mercado (Unity).

O foco dos resultados está em calcular o tempo de trajeto, ou seja, o tempo de deslocamento de todos os agentes até o ponto seguro. Não será representada a total lógica de uma evacuação em plataformas, que inclui outros aspectos como contagem de pessoal, resgate dos profissionais, dentre outros procedimentos.

## III. ARQUITETURA

Com o 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. Por isso, foi escolhida a engine de jogos Unity [6] como ecossistema deste trabalho.

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 [7], arquitetura e construção, configuração de produtos e simulações médicas [8] e de segurança [9] [10].

#### A. 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 (um deles é mostrado na Figura 1). Isso afetará na qualidade dos resultados, podendo não refletir os tempos de uma evacuação em casos reais. Por outro lado, computacionalmente, modelos mais simples tendem a ter uma performance de renderização melhor. É importante, portanto, 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.



Figura 1. Modelo de plataforma tipo FPSO

#### B. Malha de Navegação

Para que os agentes consigam caminhar no modelo importado, o Unity permite a criação de uma malha de navegação (Navigation Meshes – NavMesh) – Figura 2. 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.

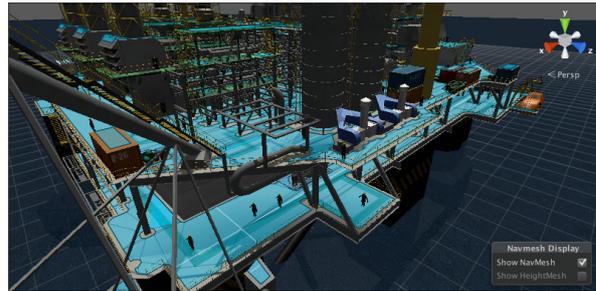


Figura 2. Malha de navegação

#### C. Agentes

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.

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

## IV. 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.

#### A. 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 do todo, dos caminhos, dos gargalos e pontos de melhoria nas estruturas.

Cada local 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, como é o caso das plataformas de petróleo.

Portanto, neste trabalho foram criados três tipos de câmeras. 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.

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. 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 (Figura 3). Quanto maior o detalhe da cena capturada por cada câmera, pior será a performance.

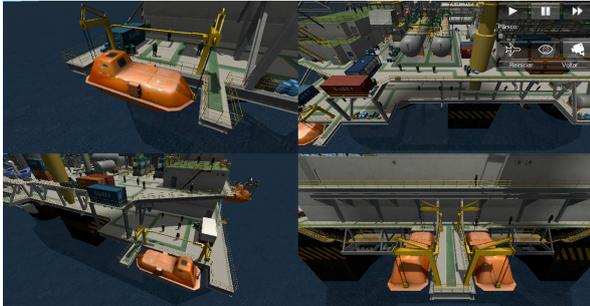


Figura 3. Visão de vigilância com quatro câmeras

A principal câmera, no entanto, é a 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 aumentadas 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.

### B. Agentes

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.

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.

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.

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.

### C. 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.

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 de pré-processamento, 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. 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.

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, e comparar com resultados reais, de acordo com as normas de segurança e com os resultados obtidos por simulações com pessoas.

## V. 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.

#### A. Cenário Simulado

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

TABELA I. CONFIGURAÇÃO DOS CENÁRIOS

Plataforma	Pessoas	Baleeiras
FPSO 1	200	4
FPSO 2	80	2
Semissubmersível	160	4

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.

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 gerar conflitos de rotas, como por exemplo, alguns agentes usando uma mesma escada (Figura 4) para subir ou para descer ou engarrafamentos em corredores estreitos.

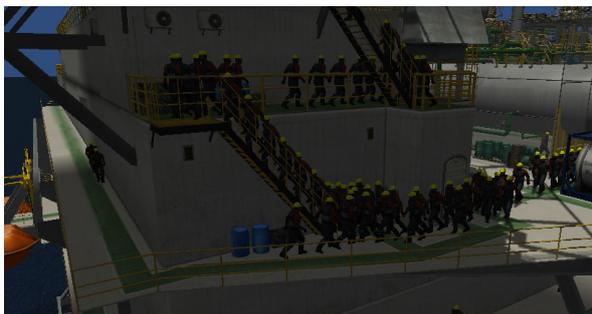


Figura 4. Filas nas escadas

Em um cenário de vazamento de gás, todos os agentes se encaminharão para o único heliponto da plataforma (Figura 5), percorrendo o caminho mais curto do seu local de trabalho até o mesmo.



Figura 5. Agentes chegando ao heliponto

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.

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.

#### B. Tempos Registrados

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

TABELA II. TEMPOS DE DESLOCAMENTO

Plataforma	Gás	Fogo
FPSO 1	06:13	05:33
FPSO 2	04:25	04:43
Semissubmersível	04:16	04:11

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 a FPSO 1, 10:00 para a FPSO 2 e 12:30 para a Semissubmersível.

Os tempos registrados no sistema são menores do que os tempos reais 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. 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 salva-vidas) 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 atrás.

No entanto, todas essas etapas possuem tempos padrões, somando mais ou menos um total de 8 minutos: acionamento do alarme (1 min), percepção (2 min) e contagem das pessoas (5 minutos). O sistema desenvolvido é essencialmente para calcular o tempo de locomoção na plataforma, ou seja, o tempo entre essas etapas. E são exatamente esses tempos que 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. Se somarmos o tempo dessas etapas aos resultados obtidos na Tabela II, percebe-se a proximidade com os tempos obtidos nos simulados com pessoas.

Nas simulações realizadas, comportamentos semelhantes ao de pedestres foram observados nos agentes, como retratado em [11] e [12]. Grandes filas foram formadas resultando em aglomerados em portas, escadas e corredores. O comportamento homogêneo defendido em Emergency Crowd Simulation for Outdoor Environments [13] também foi alcançado, demonstrando ações coletivas e não individuais.

As médias apresentadas também não englobam o abandono, ou seja, o tempo necessário para que as baleeiras sejam lançadas ao mar.

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.

O sistema permite o aumento de velocidade dos agentes (Figura 6), porém esse tipo de atitude não deve ser tomada a fim de evitar o pânico. Claro que em situação extremas, as pessoas se comportam de forma imprevisível, podendo ocorrer esse tipo de situação.

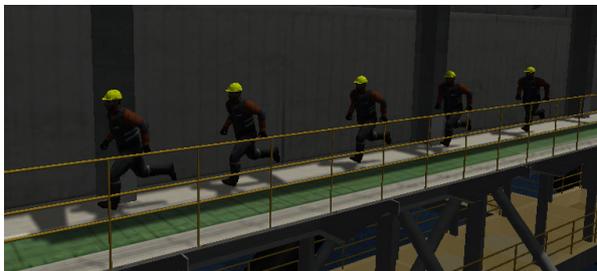


Figura 6. Agentes correndo

No entanto, realizar simulações somente com agentes correndo ou com alteração de velocidade não são válidas porque esses comportamentos sozinhos não representam a realidade de uma evacuação desordenada.

### C. Desempenho

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 III 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.

TABELA III. DESEMPENHO DO SISTEMA

Plataforma	Nº de Triângulos	Câmera Central	Visão Primeira Pessoa
FPSO 1	3 milhões	24 fps	20 fps
FPSO 2	1,4 milhões	30 fps	50 fps
Semissubmersível	4 milhões	9 fps	15 fps

Os resultados alcançados permitiram a execução de todos os cenários em tempo real, principalmente nas FPSOs. 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 (Tabela IV). Neste sistema, entende-se pré-processamento pelo tempo necessário para que as simulações estejam aptas para iniciar.

TABELA IV. TEMPOS DE PRÉ-PROCESSAMENTO EM SEGUNDOS

Plataforma	Gás	Fogo
FPSO 1	83,67	55,30
FPSO 2	68,42	30,81
Semissubmersível	222,31	178,11

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 porque os caminhos até as baleeiras (Figura 7) 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.

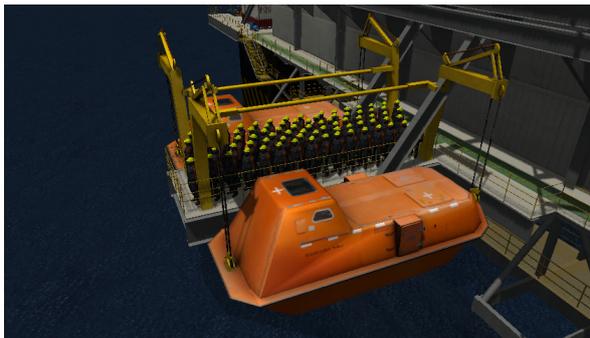


Figura 7. Agentes no fim da simulação de incêndio

## VI. CONCLUSÃO E TRABALHOS FUTUROS

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.

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.

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 [12] e [14] podem aproximar o trabalho desenvolvido à realidade.

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.

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 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.

## REFERÊNCIAS

- [1] S. R. Musse, and D. Thalmann, 2007. Crowd Simulation. Springer-Verlag.
- [2] Massive Software. Massive (Multiple Agent Simulation System in Virtual Environment). Disponível em: <http://www.massivesoftware.com>
- [3] Oasys Software. MASSMOTION. Disponível em: <http://www.oasys-software.com>
- [4] F. Tecchia, C. Loscos, and Y. Chrysanthou, 2002. Visualizing Crowds in Real-Time. Computer Graphics Forum, 21(4): 753-765.
- [5] S. R. Musse, V. J. Cassol, R. A. Rodrigues et al., 2012. CrowdSim: Uma ferramenta desenvolvida para Simulação de Multidões. XI SBGames.
- [6] Unity Technologies. Unity. Disponível em: <http://unity3d.com>
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Ribeiro, J. E. Almeida, R. J. F. Rossetti, A. Coelho, A. L. Coelho, 2012. Using Serious Games to Train Evacuation

- Behaviour. 7th Iberian Conference on Information Systems and Technologies (CISTI)
- [11] D. Helnbing, P. Molnar, 1998. Self-Organization Phenomena in Pedestrian Crowds. ArXiv Condensed Matter e-prints (June).
- [12] L. Luo, S. Zhou et al. 2008. Agent-based Human Behavior Modeling for Crowd Simulation. Computer Animation and Virtual Worlds 19(3-4): 271-281.
- [13] O. Oğuz, A. Akaydin, T. Yılmaz, and U. Güdükbay, (2010). Emergency crowd simulation for outdoor environments. Computers & Graphics, 34(2): 136-144.
- [14] N. Pelechano, N. I. Badler, 2006. Improving the realism of agent movement for high density crowd simulation. University of Pennsylvania, Center for Human Modeling and Simulation.