Universidade de São Paulo
Instituto De Ciências Matemáticas e de Computação

# Relatório Final:

# Implementação de um Sistema de Busca e Resgate Baseado em Inteligência de Grandes Enxames Acelerados em Hardware

Matheus Luis Oliveira da Silva 1
Prof. Dr. Eduardo do Valle Simões (Orientador) 2

Programa Unificado de Bolsas de Estudo Para Apoio e Formação de Estudantes de Graduação — PUB/USP

**Conteúdo:**


**Introdução** **2**

**Objetivos** **3**

**Decisões relativas ao fluxo do projeto** **4**

**Desenvolvimento do Projeto** **5**

4.1 Arquitetura do software 5

4.2 Ferramenta desenvolvida 7

4.2.1 Funcionamento e visualização do swarm 9

4.2.3 Back End 10

Interface com Usuário 10

Componentes Gráficas e pipeline 11

Comportamento das formigas 12

**Resultados** **13**

Número de agentes simulados simultaneamente suficientemente grande, mesmo sem aceleração gráfica, até o momento: 13

Dois tipos de feromônio mostraram-se eficazes: 14

Rotas encontradas sem coordenadas previamente conhecidas 14

Parceria com laboratório de pesquisa biológica da UFTM 15

**Conclusão** **15**

**Referências** **17**

# 1. Introdução

As técnicas de busca e resgate que contam com cães, câmeras móveis, sonares e radares, ainda são utilizadas para encontrar pessoas desaparecidas, corpos em desastres naturais, localização aérea de objetos de interesse, entre outras. Porém, com o desenvolvimento tecnológico de drones e robôs autônomos, as técnicas de busca e resgate atualmente vem sendo atualizadas (Ko & Lau, n.d.) e, para além da utilização de poucos robôs nesse campo, introduz-se a tecnologia dos Sistemas de Inteligência de Enxame, que abre espaço para novas abordagens no campo da robótica.

A chamada Inteligência de Enxame (IE), ou Swarm Intelligence, foi abordada por Craig W. Reynolds, em 1987, em seu artigo “Flocks, Herds, and Schools: A Distributed Behavioral Model” (Reynolds, 1987) no qual modelou um programa de vida artificial que ele chamara de “Boids”, termo proveniente do comportamento em bando dos pássaros. Esse sistema iniciou o estudo que atualmente se tornou um campo da inteligência artificial que estuda sistemas descentralizados e auto-organizados pelo comportamento coletivo de seus elementos. Tal tecnologia é utilizada em problemas nos quais há uma grande quantidade de elementos a serem trabalhados, os quais possuem algum tipo de interconexão e soluções que exigem dinamicidade, como, por exemplo: um grande número de drones voando simultaneamente, como estudado em (Majd et al., 2018); ou um grande número de pequenos robôs terrestres que se movem de maneira auto-organizada (Rubenstein et al., 2014), Nestas abordagens são utilizadas tecnologias bioinspiradas, aplicações que adaptam processos biológicos para soluções computacionais, como em aplicações de colônias de formigas (Dorigo et al., 2006), em aplicações inspiradas no comportamento de abelhas (Yang et al., 2007), ou em otimização via enxame de partículas (Kennedy & Eberhart, 1995), por exemplo.

Ao se trabalhar com dezenas, ou até centenas de pequenos robôs físicos, entretanto, acabam surgindo empecilhos operacionais, como por exemplo, o manejo de uma grande quantidade de componentes eletrônicos, carregamento da bateria de cada um dos agentes do enxame, posicionamento e inicialização manual de cada robô no ambiente de testes ou atuação, entre outros problemas que podem surgir quando se trata de um grande volume de robôs autônomos. Assim, trabalhar em uma simulação para estudar Enxames pode ser uma boa opção, a qual abstrai a complexidade de lidar com robôs físicos, possibilitando a criação de simulações das mais variadas aplicações de enxames, como sistemas inspirados em abelhas (Lim et al., 2021),de comportamento de robôs autônomos em grupo, junto a outras Inteligências Artificiais (IAs) como em (Hiejima et al., 2019) ou até mesmo em aplicações de caça de alvos ou invasores, nas áreas militar e de segurança (Majd et al., 2018).

Todavia, mesmo em trabalhos com simulações, quando se lida com muitos agentes simultâneos, um grande processamento computacional é exigido, o que pode acarretar em problemas de performance, limitando a atuação do enxame devido ao limite do número de indivíduos. É nesse contexto que é proposta essa pesquisa, que propõe o desenvolvimento de um sistema de Inteligência de Enxame acelerado em hardware, baseado em colônia de formigas, para estudar a eficiência e a performance do sistema em relação às variações possíveis dos parâmetros, como a influência do número de agentes, quantidade de objetos a serem coletados em uma aplicação de busca e resgate. Com a aceleração em hardware, espera-se que uma grande quantidades de agentes possam estar presentes simultaneamente na simulação, sem que haja perda de performance do sistema, o que pode potencializar a ação do Enxame e possibilitar testes diversos, variando a quantidade de agentes e estudando suas interações.

# 2. Objetivos

Propõe-se que um sistema com aceleração em hardware possa agregar novas abordagens para os sistemas de enxame, visto que, devido ao processamento paralelo das GPUs, uma quantidade maior de agentes poderá ser simulada simultaneamente, quando comparado a um sistema convencional limitado pelo processamento da CPU, o que pode acelerar e viabilizar possíveis testes em simulação, que antes demandariam muito tempo, energia e infraestrutura no ambiente real, com robôs físicos. Assim, esse trabalho visa avaliar se a aceleração em hardware irá melhorar a velocidade de execução do sistema, possibilitando tanto estudos a respeito da importância da quantidade de agentes em simulação, quanto a modelagem de robôs com comportamentos complexos e em ambientes dinâmicos.

Com isso, esse projeto tem como objetivo o desenvolvimento de um sistema que simulará as ações de um formigueiro (da Silva et al., 2005) nas seguintes tarefas: busca por recursos,exploração de território, otimização de rotas através de deposição de feromônio e coleta de objetos. O sistema de controle terá como base teórica a inteligência de enxames, com a implementação de centenas, ou até milhares de agentes programáveis.

Os agentes do sistema serão adaptações artificiais de formigas, as quais se moverão num ambiente terrestre e terão a sua estigmergia, ou seja, seu sistema de comunicação a base de feromônios, estudada e adaptada lógica e graficamente para a simulação (da Silva et al., 2005). Essa comunicação será implementada como uma matriz de deposição de feromônio, representada por uma matriz de pixels que serão percebidos artificialmente pelas antenas das formigas que se sobrepuserem a eles.

O processamento da sobreposição das formigas com a matriz de feromônio talvez seja o processo mais custoso de todo o sistema, visto que a matriz deve ser atualizada e acessada por cada formiga a cada iteração do looping principal da simulação, portanto, para que não haja uma queda de desempenho com o aumento do número de formigas em simulação, todo o processamento dessa função deverá ser passada para a GPU, a fim de otimizar e amplificar a capacidade de processamento da simulação.

A parte visual, com a representação gráfica na tela, será implementada em OpenGL, e a conexão com a placa de vídeo para o processamento lógico será feita através da API desenvolvida pela Nvidia CUDA.

Além disso, propõe-se a implementação de um algoritmo evolutivo para otimizar os parâmetros operacionais das formigas, otimizando assim, todo o comportamento e autorregulação do enxame, que deverá funcionar com uma grande quantidade de indivíduos, o que fará com que o algoritmo evolutivo também deva ter o processamento mais denso realizado em placa gráfica.

Para isso, uma pesquisa bibliográfica deverá ser feita, tanto para conhecer o funcionamento biológico da inteligência por trás do formigueiro, como das implementações e abordagens já feitas. Após isso, será realizada a integração das tecnologias de processamento gráfico e aceleração das ferramentas, para, então, implementar a simulação, sempre buscando um bom desempenho, tanto da parte gráfica, quanto da parte lógica da simulação.

A princípio, o foco é que o sistema otimize rotas para determinados objetos e adapte-as a diferentes configurações do ambiente que poderá ser modificado dinamicamente, fazendo com que os agentes do enxame alcancem os objetivos e retornem

ao seu local de origem, coletando os objetos. Além disso, futuramente podem ser aplicadas novas ideias para a abordagem da inteligência de colônia de formigas, como testar o comportamento de duas colônias cooperando ou competindo, ou então combinar componentes de outros sistemas IE, como o inspirado no comportamento das abelhas.

# 3. Decisões relativas ao fluxo do projeto

No decorrer do desenvolvimento do projeto, algumas decisões foram tomadas quanto ao objetivo final da ferramenta, bem como do primeiro produto que seria finalizado até a entrega desse projeto. Assim, o escopo da ferramenta final foi ampliado para atender qualquer tipo de swarm, não só o de formigas.

Com o novo objetivo, as atividades desenvolvidas nesse projeto foram destinadas a preparar uma primeira versão da ferramenta que suporte o swarm de formigas, como antes planejado, mas que fosse otimizada e modularizada o suficiente para comportar novos swarms de forma prática, além de comportar a implementação de um algoritmo evolutivo de maneira integrada com o setup das simulações.

Dessa forma, segue abaixo uma tabela que lista algumas das alterações feitas nas atividades previstas na elaboração do projeto:

| **Atividade Original** | **Nova atividade** |
|---|---|
| Revisão bibliográfica sobre Inteligência de Enxame e Algoritmos Evolutivos. | Sem alterações |
| Estudo do comportamento do sistema de enxame dos formigueiros. | Sem alterações |
| Estudo da API de computação Gráfica OpenGL para a parte gráfica da simulação, bem como a otimização desta em GPU, buscando funções otimizadas e eficientes para grandes simulações. | Sem alterações |
| Estudo da API CUDA de aceleração de hardware, para a realização dos cálculos necessários para a simulação da população de agentes de simulação. | Estudo feito, porém, no remanejo das prioridades do projeto, a API CUDA não foi implementada, bem como a aceleração em hardware proposta por ela. |
| Projeto da arquitetura e desenvolvimento do sistema que integrará a lógica do formigueiro, a parte gráfica e a aceleração de hardware. | Aceleração OpenGL + CUDA foi substituída pela API OpenGL. Além da projeção do sistema para comportar, não somente formigas, mas qualquer tipo de agente. |
| Implementação de um Algoritmo evolutivo para otimização da eficiência das formigas | Com o avanço da aplicação gráfica, bem como da projeção da ferramenta para |

| | |
|---|---|
| e da população do formigueiro como um todo. | comportar simulações diversas e otimizadas, a feature do algoritmo evolutivo foi prevista, com a ferramenta tendo sua estrutura preparada para receber o AE futuramente. Além disso, não somente um AE será comportado, mas sim uma estrutura de implementação para qualquer tipo de AE via interface interativa com o usuário. |
| Integração de todas as tecnologias estudadas e desenvolvidas, para desenvolvimento do sistema final. | Sem alterações. |
| Realização de testes de desempenho do enxame em diferentes cenários e situações, análises de melhorias, assim como futuras aplicações e aperfeiçoamentos. | Testes realizados, bem como melhorias constantes na arquitetura da ferramenta e seu back-end.<br>No estágio atual a ferramenta está pronta para receber diversas novas features já planejadas. |

Tabela 1: Descrições das alterações feitas nos objetivos originais do projeto.

# 4. Desenvolvimento do Projeto

Como parte do projeto, o aluno teve de atualizar a revisão bibliográfica do projeto, para embasar as decisões de desenvolvimento tomadas e definir os objetivos finais da ferramenta. Seguida a revisão, a arquitetura da ferramenta foi esquematizada e, a partir dela, as funcionalidades implementadas, desde o gerenciamento de memória e comunicação com a GPU, até a interface gráfica de interação com o usuário.

Como objetivo final desse primeiro projeto, a primeira versão da ferramenta deveria ser disponibilizada à comunidade para acompanhamento e contribuições Open-source. Ainda que não finalizada a versão beta, uma primeira build esta disponível no GitHub do aluno idealizador e programador da ferramenta, disponível em:

https://github.com/matheuslosilva/Hardware-Accelerated-Ant-Colony-Based-Swarm-System

Dessa forma, segue abaixo todo o caminho até o resultado do presente projeto.

## 4.1 Arquitetura do software

Segue abaixo na Figura 1 um diagrama simplificado da ferramenta criada, com uma estrutura top down, com dependências verticais, de cima para baixo:

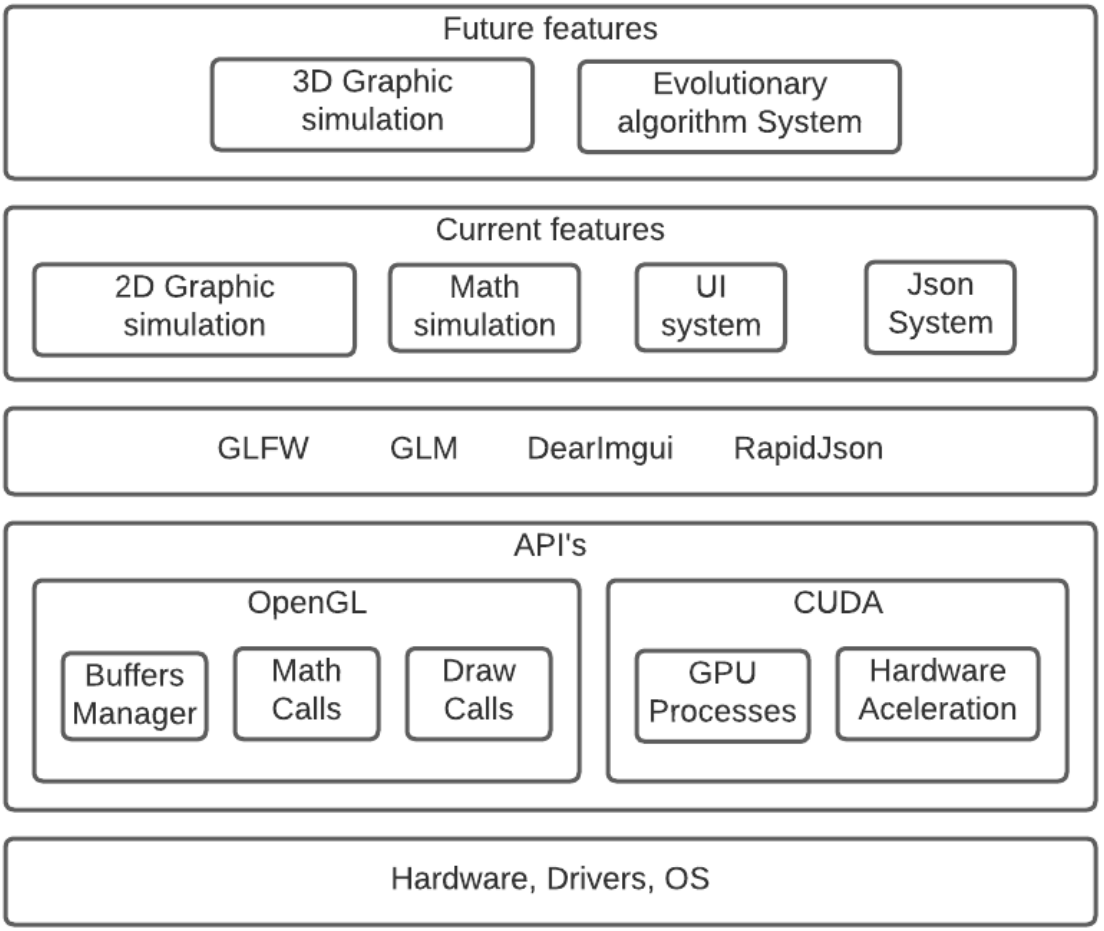


Figura 1: Estrutura top-down das dependências da ferramenta, bem como suas features principais.

Na imagem vemos que a base do sistema tem quatro principais dependências: o GLFW (glfw [21]), uma API de criação e gerenciamento de telas, superfícies e tratamento de inputs do usuário, geralmente usada com as API's de desenvolvimento gráficas OpenGl, Vulkan, ou similares; o GLM, que fornece uma biblioteca de funções e componentes matemáticas para as operações com os buffers de dados das componentes gráficas; o DearImgui (dearimgui[22]), uma biblioteca open-source para a criação de interfaces gráficas para C++; e, por fim, o rapidJSON, uma API para C++ para manipular arquivos JSON.

Logo acima, temos as duas API's centrais do sistema, a primeira, OpenGL, para a parte gráfica da ferramenta, bem como o gerenciamento dos buffers de dados e abstração da comunicação com os drivers de vídeo, e a segunda, CUDA (cuda [23]), uma API da Nvidia para aceleração gráfica com processamento em GPU. Espera-se que esse conjunto promoverá à ferramenta um back-end moderno e eficiente de processamento gráfico, conferindo à ferramenta a velocidade necessária para as simulações futuras.

Como as principais features já em desenvolvimento, tem-se:

- 2D Graphic simulation: uma simulação em 2D dos agentes em movimento e interação com o ambiente, ninho e objeto de interesse.
- Math simulation: toda a parte matemática da simulação funcionando de maneira independente da parte gráfica, para que seja possível aumentar

ainda mais a eficiência da simulação desligando a parte visual sem que a simulação pare de avançar.

- UI system: uma interface com o usuário para que gerencie a simulação e seus componentes, ainda rudimentar, porém funcional.
- JSON System: um início do que será a parte de armazenamento dos dados da simulação bem como as configurações iniciais e intermediária das simulações, todas em arquivos JSON.

Além das supracitadas, há uma projeção para as próximas features da ferramenta, que vão conferir robustez e complexidade à mesma, que são simulações em ambientes de 3 dimensões e a implementação e integração de um gerenciador de algoritmos evolutivos aplicados aos swarms simulados.

## 4.2 Ferramenta desenvolvida

A ferramenta proposta tem como objetivo fornecer ao usuário uma plataforma de simulação gráfica capaz de definir as configurações iniciais do ambiente e dos agentes do swarm. Assim, para além das aplicações encontradas na bibliografia estudada, a ferramenta traz uma inovação para o estado da arte: o foco na parte gráfica, ou seja, na possibilidade de visualização em tempo real do swarm agindo na solução do problema.

Dessa forma, pode-se observar, na Figura 2, a tela principal da ferramenta, com uma simulação em curso. Nessa imagem, a esquerda temos a interface com o usuário, Figura 3, com sliders para definição do posicionamento da colônia de formigas e das fontes de alimento, bem como a quantidade de formigas em cada colônia, e, logo acima, temos o controle de fluxo da ferramenta, para iniciar, pausar, parar e fechar a simulação.

Além disso, é claro, tem-se a direita a tela da simulação, com as formigas sendo visualizadas à distância (nota-se apenas pontos brancos nessa imagem), a colônia no centro em marrom, as fontes de alimento em verde e as trilhas de feromônio em verde e vermelho.

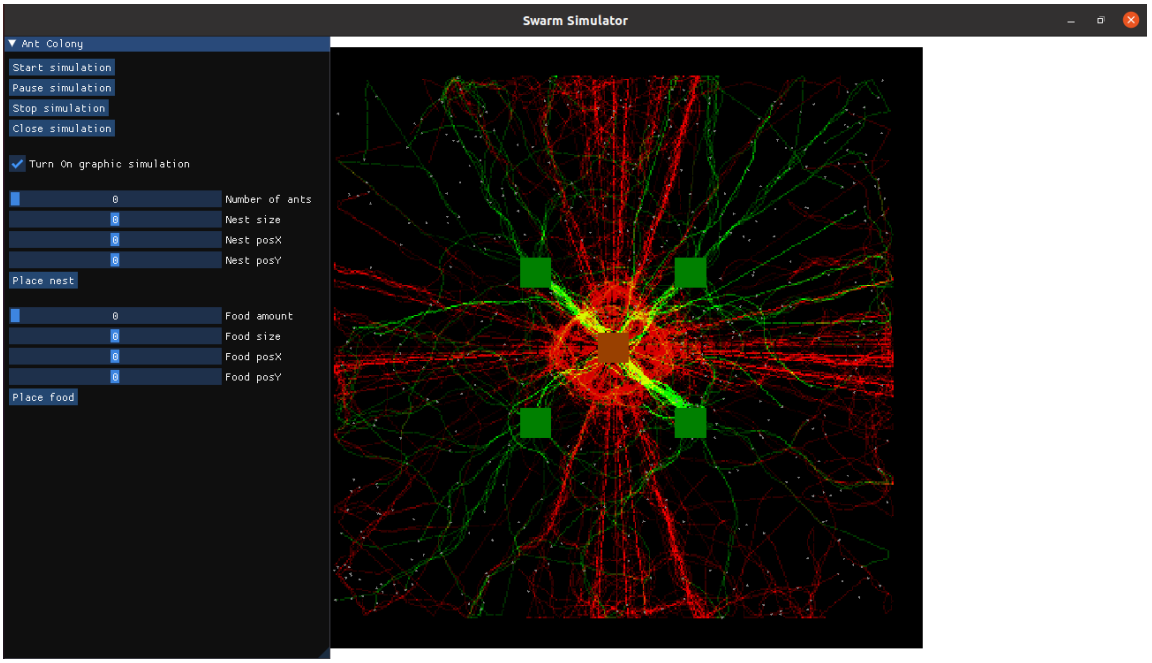


Figura 2: Tela principal da ferramenta com uma simulação em curso.

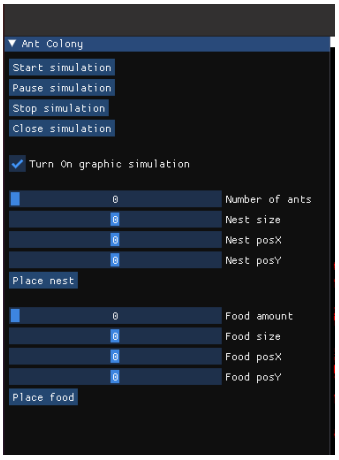


Figura 3: Interface com usuário, botões de controle da simulação e sliders para definição dos valores.

A princípio, essa interface mostrou-se suficiente para os testes com a ferramenta, entretanto, o uso dos sliders e botões para posicionamento inicial das componentes não se mostrou prático. Assim, na nova etapa de desenvolvimento da ferramenta uma nova interface está sendo criada, que pode ser visualizada na Figura 4, com novas funcionalidades, como:

- Posicionamento das componentes da ferramenta com o mouse;
- Importação de um arquivo JSON (botão “+” na aba Experiments, na Figura 4) com as informações das configurações iniciais da simulação, configurando tudo automaticamente após a importação;
- Interações em tempo de execução na simulação;
- Ligar e desligar a parte gráfica da simulação;
- Aba para o gerenciador do algoritmo evolutivo que será implementado no swarm;
- Aba para log de todas as informações da simulação, swarm, Algoritmo Evolutivo, etc.

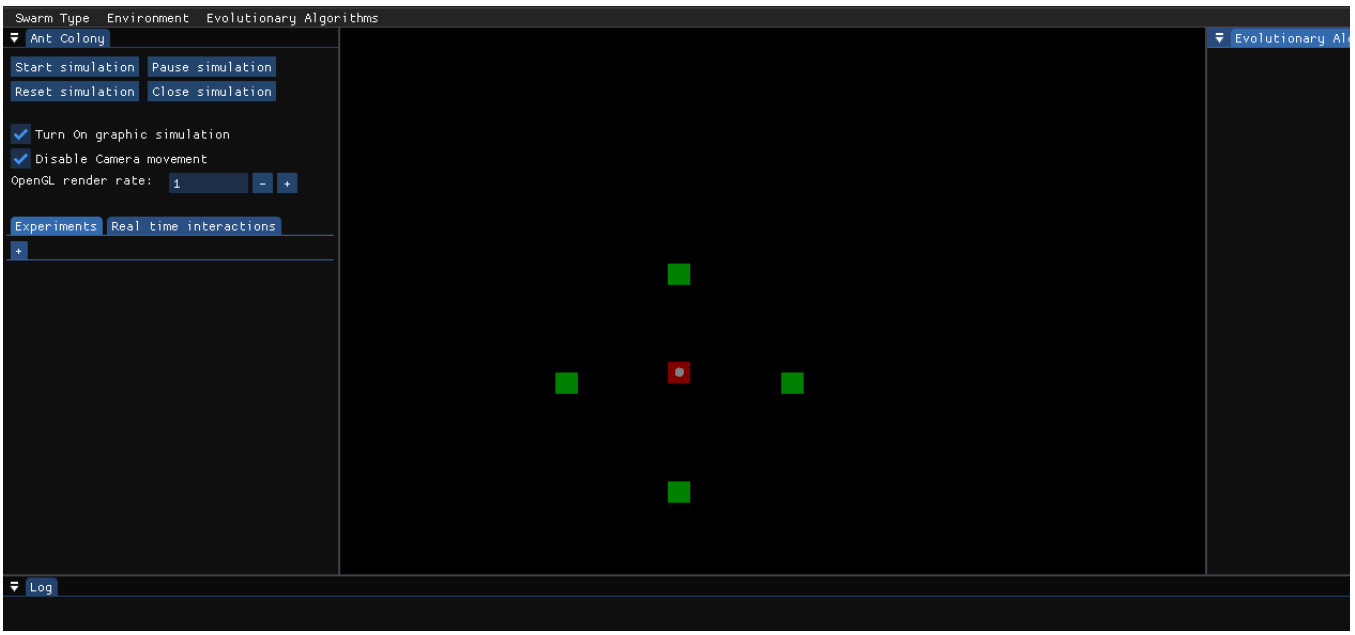


Figura 4: Nova interface com o usuário, em desenvolvimento, com aba de componentes, aba de log e aba de configuração dos algoritmos evolutivos.

## 4.2.1 Funcionamento e visualização do swarm

Além das colônias e das fontes de comida já citadas, é possível observar também os agentes do swarm em movimento durante todo o processamento da simulação. No caso dessa ferramenta, o swarm de formigas, denominadas como agentes, foi representado por um conjunto de triângulos cinzas, polígono mais simples possível a fim de otimizar a açãoda GPU, que se movimentam pelo ambiente e interagem com os elementos presentes. A interação dos agentes com o ambiente segue os seguintes princípios:

1. Toda a superfície da simulação possui uma malha de pixels, que pode ser acessada e ter os componentes RGBA (conjunto de 4 valores que variam de 0 a 255 e representam, respectivamente, a proporção de vermelho, verde, azul e transparência do pixel, que, quando renderizado e conferido ao elemento gráfico, no caso o pixel, conferem a ele uma cor específica) de cada pixel alterada.

2. Na natureza, as formigas se utilizam da deposição de feromônio no solo onde passam, deixando um rastro que pode ser identificado por outras formigas. Nessa simulação, o feromônio foi modelado por componentes RGBA de um pixel, ou seja, por onde as formigas passam, elas alteram a componente RGBA de cada pixel que compõe a sua trajetória. Assim, isso possibilita que outras formigas passem por esse local e verifiquem as componentes RGB dos pixels ao redor, podendo tomar decisões baseadas na quantidade de RGB depositado naquele local. A figura 5 representa esquematicamente um modelo da formiga, triângulo, e dos sensores simulados que lêem os valores RGB dos pixels da malha.

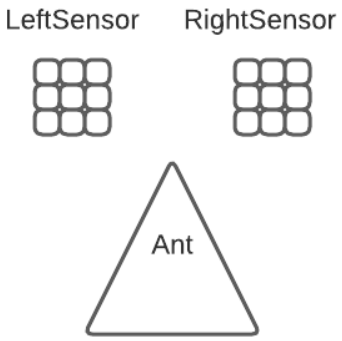


Figura 5: Representação de uma possível formiga na simulação

3. Na figura em específico, essa formiga possui um sensor 3x3, ou seja, lê um quadrado de 9 pixels, faz a média dos valores RGB, da direita e da esquerda, e toma uma decisão, virando para a direita ou para a esquerda, baseada na diferença da quantidade de feromônio de um lado e de outro.

4. Com a malha de pixels, então, é possível que o usuário também enxergue as trilhas de feromônio, uma vez que as componentes RGBA da matriz de pixels se convertem como cores visíveis na placa de vídeo, o que pode ser observado na figura 6.

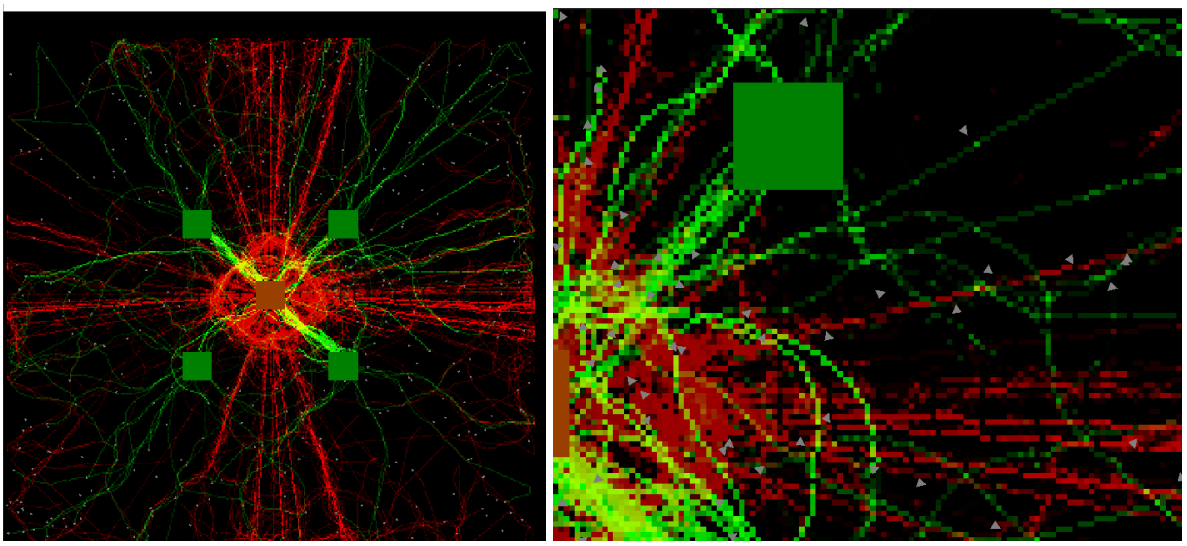

Figura 6: Prints de uma simulação com 4 fontes de alimento à esquerda e recorte com zoom à direita.

5. Além de ter acesso a malha de pixels, os agentes também verificam colisão com as colônias e as fontes de alimento, tomando decisões baseadas nesses eventos também.

Com simples instruções, o swarm possui um comportamento complexo, como se pode observar na Figura 6, com padrões de comportamento inesperados, promovidos pela interação dos agentes do swarm, que, com apenas dois tipos de feromônio, nesse caso ilustrado, foram capazes de traçar algumas trilhas, em verde, da colônia até as fontes de alimento, como visto na parte esquerda da figura.

## 4.2.3 Back End

A princípio, o projeto não previa o conhecimento prévio nem a predefinição de nenhuma tecnologia ou método de desenvolvimento específicos. Desse modo, o aluno teve liberdade de investigar e escolher o melhor conjunto favorável de ferramentas para desenvolver um simulador gráfico. Com isso, foram definidas algumas etapas de desenvolvimento e as características principais da ferramenta, explicadas com detalhes nas seções a seguir.

- **Interface com Usuário**

Já mostrada anteriormente na seção 4.2, a interface gráfica se utilizou de uma biblioteca chamada ImGui, que funciona sem nenhuma dependência externa, por meio da abstração de toda a renderização de buffers, viabilizando ao desenvolvedor uma interface de desenvolvimento ágil e confortável para criação de interfaces gráficas de criação, desenvolvimento, debug etc. Com essa biblioteca a interface da ferramenta foi desenvolvida com abas, botões, campos de preenchimento, checkboxes, telas de log de simulação, etc.

Assim, o objetivo final é que o pesquisador que virá a usar a ferramenta não precise alterar o código em nenhum momento, ou ao menos minimizar adaptações, quando for criar sua própria simulação ou experimento, realizando tudo pela interface e funções pré disponibilizadas a ele.

## ● Componentes Gráficas e pipeline

Como já abordado nas seções anteriores, a API OpenGL foi a utilizada para todo o pipeline gráfico da ferramenta. A princípio, o desenvolvimento focou na implementação das formigas e, para isso, componentes simples foram suficientes para representação das componentes, contando com triângulos para a representação das formigas e quadrados para a representação das colônias e fontes de alimento. Com a API gráfica a definição das primitivas são de fácil manipulação, com rotinas de desenho, alocação de memória gráfica e de estruturas de dados já prontas, facilitando a comunicação com a placa gráfica.

O maior desafio encontrado foi a representação do feromônio depositado pelas formigas, uma vez que elas devem acrescentar feromônio por onde passam bem como detectar o feromônio que já está ali, depositado pelas outras agentes, a fim de tomar uma decisão de rota com base na quantidade do mesmo.

Como a APi não prevê o uso da biblioteca para tracejamento de linhas simples na tela, mas sim de formas mais complexas, texturas e etc, foi preciso encontrar uma alternativa para que as formigas conseguissem traçar um caminho de feromônio. Assim, foi proposta uma matriz de números de 32 bits, que representam as cores dos pixels que compõem a textura de feromônios, conforme mostra a Figura 7 abaixo:

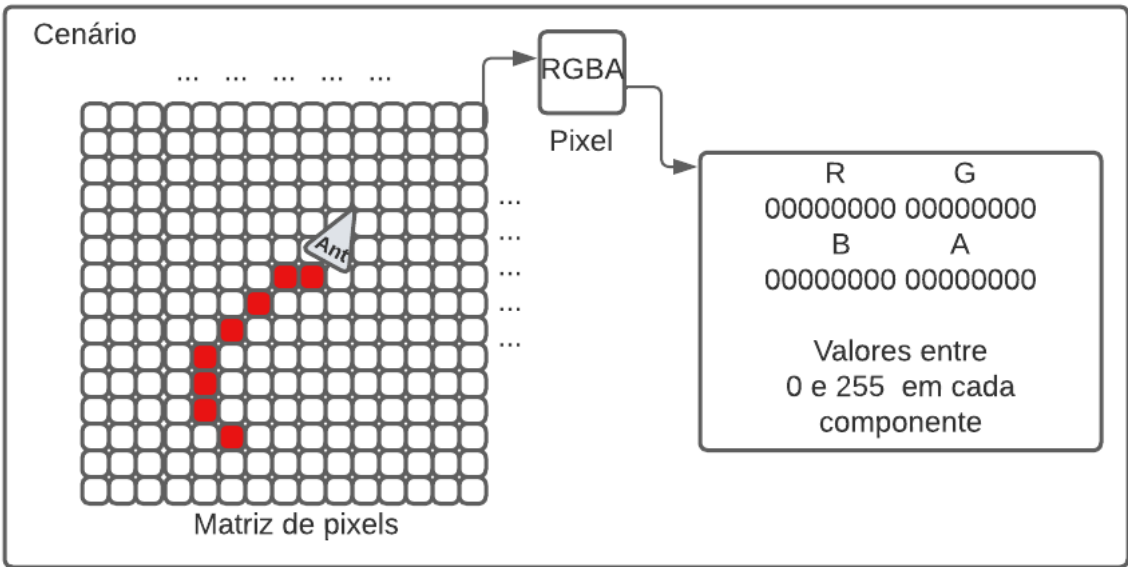


Figura 7: Representação de um recorte da textura de pixels, no caso de 15X15 pixels, com a representação individual do valor de 32 bits que compõem a cor do pixel representado.

Dessa forma, é possível projetar a posição da formiga na tela e convertê-la para uma posição da matriz de pixel e, assim, “depositar feromônio” na forma de cor na textura, adicionando valores de 1 a 255 no valor RGBA do pixel. No caso foram usados quatro canais diferentes para que fosse possível representar até 4 feromônios diferentes, um em cada canal (vermelho, verde, azul e transparência), com cada formiga podendo acessar um pixel específico e alterar a cor do mesmo, como na imagem, com o caminho da formiga sendo pintado de vermelho.

## ● Comportamento das formigas

A representação da formiga foi criada a partir da observação da natureza e da ajuda do biólogo Rhainer Guillermo do Lestes Lab da universidade federal do triângulo mineiro (UFTM), especialista em insetos, que orientou a parte biológica da pesquisa, auxiliando na adaptação da formiga para a simulação.

Como mostrado anteriormente, então, a formiga baseia seu comportamento inteiramente pelo feromônio detectado no chão. Dessa forma, seus sensores da direita e da esquerda detectam uma amostra de feromônio de cada lado e, a partir delas, decide para qual lado virar e seguir seu caminho. Em suma, dependendo do lado que mostrar mais feromônio depositado, a formiga opta por esse lado ou não, se aproximando ou não de um determinado tipo de feromônio.

Com base nesse comportamento, criou-se um agente com os seguintes estados de ação:

- Explorer: A formiga exploradora, que se afasta do feromônio vermelho para forrageamento (exploração do ambiente a procura do objeto de interesse.
- Carrier: A formiga descobridora, que se aproxima do feromônio vermelho, carregando a comida que acabara de encontrar da fonte para o ninho. Ela, por sua vez, deposita feromônio verde.
- NestCarrier: A formiga carregadora, que já encontrou a comida e voltou pro ninho, que passa a alternar o caminho da fonte para o ninho e do ninho para a fonte. Em teoria é a formiga final, aquela que conhece o caminho alvo. Ela, por sua vez, se aproxima do feromônio verde e também o deposita, agora em maior quantidade.

A Figura 8 representa, através de um fluxograma, o processamento da formiga Explorer, indicando a passagem de estado no final do fluxo, se tornando uma formiga Carrier:

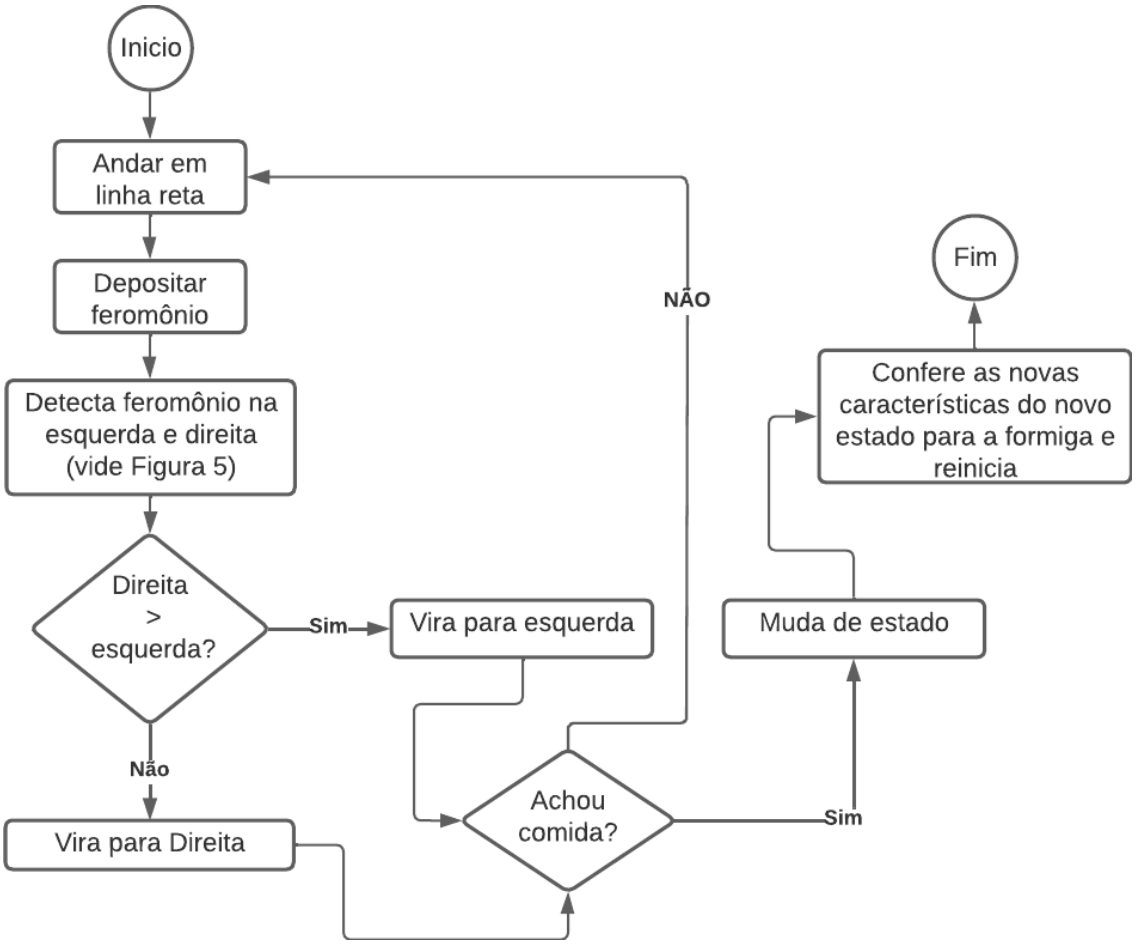


Figura 8: Fluxograma da formiga que está no estado Explorer.

# 5. Resultados

Durante o ano, o projeto caminhou muito bem, partindo de uma base já criada durante a pandemia, obteve-se avanços significativos tanto na simulação quanto na ferramenta em si (interface, funcionalidades e afins). Assim, podem ser citados os seguintes resultados mais significativos:

1) Número de agentes simulados simultaneamente suficientemente grande, mesmo sem aceleração gráfica até o momento:

O projeto implementado mostrou-se capaz de simular em 30 FPS colônias de até 300 mil formigas em um Notebook i5 sétima geração com um placa de vídeo de entrada NVidia Geforce 940mx e 8Gb de RAM. Esse resultado é bastante promissor, uma vez que a continuação do projeto prevê a aplicação da API CUDA, que será aplicada para o multiprocessamento em placa de vídeo da maioria das operações, o que, em tese, irá acelerar ainda mais a aplicação, possibilitando um número ainda maior de agentes, ou então uma velocidade ainda maior de simulação com esse número mantido.

## 2) Dois tipos de feromônio mostraram-se eficazes:

Foram mapeados dois tipos de feromônios diferentes, um para marcar o terreno já explorado (vermelho) e outro para marcar os caminhos entre a comida e o formigueiro (verde). Com esse mapeamento, as formigas foram capazes de encontrar algumas rotas e otimizar com o tempo, mostrando a eficácia do swarm com apenas dois feromônios. Além dos dois utilizados, a forma como o feromônio foi modelado comporta ainda mais dois tipos, se utilizar os outros dois canais B e A da componente RGBA, com possibilidade, ainda de explorar um novo mapeamento para suporte a um número ainda maior, abrindo espaço para comportamentos e agentes ainda mais complexos.

## 3) **Rotas encontradas sem coordenadas previamente conhecidas**

Diferentemente dos exemplos de formigueiros gráficos disponíveis na internet, que nos quais as formigas conhecem as coordenadas do ninho e dos alimentos (como se possuíssem uma visão global, top-down, de todo o ambiente, esse projeto construiu uma colönia cujos indivíduos navegam sem conhecer suas coordenadas, usando somente seu feromônio para mapear o ambiente e se localizar em relação ao ninho. Mesmo assim, a colônia é capaz de explorar o meio ambiente, encontrar objetos de interesse (comida), encontrar um caminho entre esses objetos e o ninho (base) e otimizar esse caminho para coletar a maior quantidade de objetos possível.

A Figura 9 mostra uma sequência de três fotos, em ordem cronológica, da evolução das rotas encontradas pelo swarm, o qual foi colocado inicialmente no centro de quatro fontes de alimento (quadrados verdes) e, com o tempo, traçou algumas rotas até essas fontes de alimento em feromônio verde, convergindo para uma rota para a fonte de alimento do canto superior esquerdo, conforme mostra a terceira parte da figura:

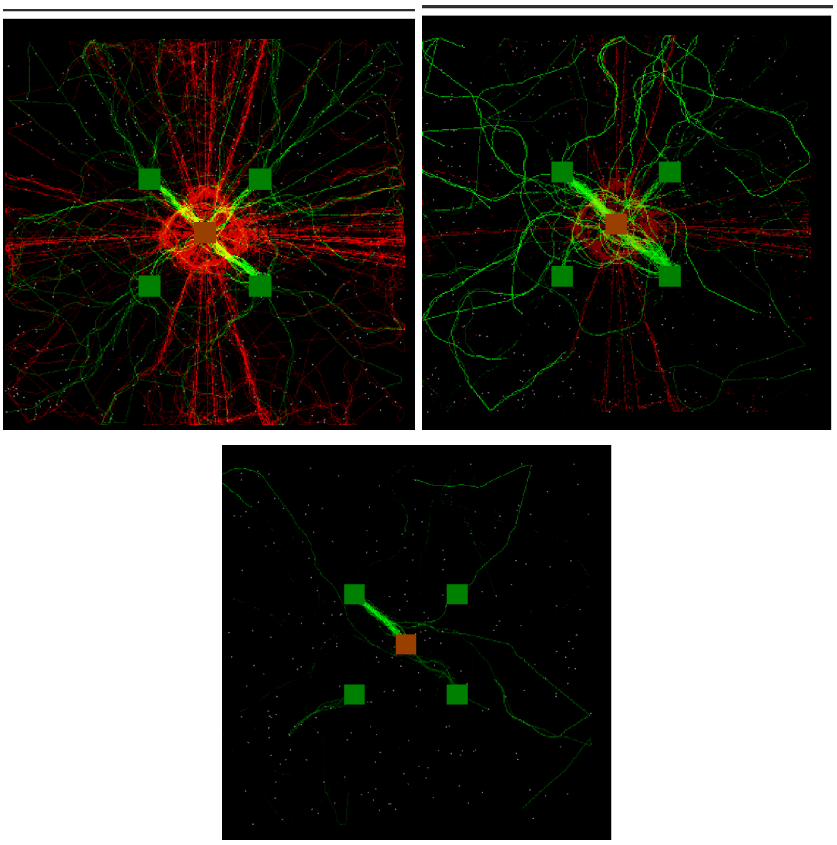

Figura 9: Sequência cronológica de prints da simulação de um swarm com quatro fontes de alimento distribuídas simetricamente

É importante ressaltar também que esse swarm foi criado pelos programadores e idealizadores da ferramenta, com a finalidade de testar a parte de simulação da mesma, para averiguar se a base do sistema seria capaz de simular um swarm com muitos agentes de maneira robusta, sem bugs, e possivelmente encontrar uma solução interessante. Deve-se dizer, então, que inúmeras outras configurações são possíveis e não foram exploradas nesta pesquisa, a qual entrega a ferramenta para qualquer pesquisador ou entusiasta que queira criar seus próprios agentes com seus próprios comportamentos, podendo obter resultados ainda mais promissores, como um swarm que consiga conectar as quatro fontes de alimento ao ninho, por exemplo.

## 4) Parceria com laboratório de pesquisa biológica da UFTM

Durante a pesquisa, a ferramenta e ideia do projeto foi apresentada para dois biólogos entomólogos, Ricardo Leite, do Instituto Federal de Piracicaba, e Rhainer Guillermo da UFTM, o qual mostrou muito interesse na pesquisa e citou diversos pontos promissores para o estado da arte.

Um dos pontos levantados por Rhainer foi o pioneirismo da ferramenta no foco do estudo das particularidades dos agentes do swarm, sem apenas focar nos objetivos finais de um dado problema (como o de otimização de rotas). Com essa possibilidade de ajuste fino das características dos agentes insetos, um pesquisador pode utilizar a ferramenta para estudar diversos comportamentos de diferentes insetos, fato esse que, segundo Rhainer e Ricardo, não é possível com as ferramentas atuais disponíveis.

Ambos citaram também a inovação na parte gráfica da simulação, com a possibilidade de visualizar em tempo real o swarm agindo no ambiente, citando a limitação

do estado da arte em simulações apenas matemáticas, sem o artifício visual que, além de fornecer um recurso poderoso para especialistas da área, facilita o entendimento dos entusiastas do comportamento do swarm. Dessa maneira, a ferramenta tem potencial educacional, além da experimentação profissional dos pesquisadores.

Espera-se que essa parceria siga durante o desenvolvimento dessa pesquisa, que continuará para além desse projeto, com objetivos que se relacionarão com a biologia, inteligência artificial, sistemas bioinspirados, entre outras áreas, que poderão usar a ferramenta para criar novos projetos interdisciplinares.

# 6. Conclusão

Dada a proposta inicial do projeto, pode-se dizer que a ferramenta desenvolvida foi além do planejado e estabeleceu novas metas ainda mais promissoras e ambiciosas, uma vez que a simulação com formigas foi feita logo nos primeiros meses do do ano, que partiu de simples triângulos se movimentando na tela para múltiplas colônias e fontes de alimento, com centenas de milhares de formigas simuladas. Com isso, foi possível traçar novos objetivos para o projeto, dado que o mínimo proposto já tinha sido realizado. Assim, a interface com o usuário pôde ser melhorada, bem como os subsistemas da ferramenta, como a preparação para a implementação da interface para algoritmos evolutivos, refatorações constantes do código, implementação de uma interface com arquivos JSON, interface com usuário melhorada, entre outras já expostas nesse documento.

Quanto à proposição de um sistema de busca e resgate, as primeiras simulações mostraram-se ineficazes, uma vez que, logo nos primeiros protótipos, o comportamento do swarm era definido por muitos parâmetros, o que torna complexa a tarefa de criar um comportamento específico para determinado problema. Mesmo assim, após diversos testes, foi possível criar swarms que cumpriam as tarefas de maneira satisfatória, o que comprova a eficácia da técnica do swarm de colônia de formigas para problemas de busca e resgate em ambientes 2D. Não obstante, dada a dificuldade de configurar um swarm manualmente, coube como próximo objetivo a integração de um algoritmo evolutivo para essa tarefa, evoluir o próprio sistema inteligente. Com essa integração, junto de todo o log de simulação com arquivos JSON organizados e modularizados, bem como a generalização para outros tipos de swarm que não os de formiga, a ferramenta se prestará ao estado da arte como um potente instrumento de ensino e pesquisa de swarm, algoritmos evolutivos e sistemas inteligentes bioinspirados.

Propõe-se, assim, a continuação dessa pesquisa com o objetivo de acrescentar à ferramenta novas funcionalidades, bem como a fundação de um laboratório de ensino e pesquisa de Swarm Intelligence e Sistemas Evolutivos no ICMC. Em adição, vale citar que esse plano já está em curso, recrutando novos alunos, professores e pesquisadores interessados na área, para aproveitar o potencial de retribuir à comunidade todo o apoio dado a essa pesquisa, com um laboratório que será sede de novas pesquisas e uma nova área de inovação para o Instituto.

O projeto continuaŕá com os novos alunos envolvidos e se ramificará em diversas áreas e objetivos em diversas frentes de projeto. Propõe-se, então, como trabalhos futuros:

1) Aceleração de toda a simulação com a implementação de multiprocessamento em GPU com CUDA.
2) Melhorias na física dos agentes: colisões e manobras mais realistas (fazer curvas como se tivesse rodinhas, não passar mais por cima da comida, do

ninho ou dos outros indivíduos, bem como, colidir com as paredes e obstáculos).

3) Simulação com ambiente 3D - As formigas poderão escalar paredes e passar por cima dos obstáculos.
4) Melhorias na interface com o usuário, com a adição de novos comandos mais amigáveis e intuitivos.
5) Inclusão do gerenciador de algoritmos evolutivos que, de forma intuitiva, se conectará com os parâmetros do swarm criado, a fim de melhorá-los para procurar por uma solução ótima para um determinado problema. Dessa forma, o pesquisador terá uma nova ferramenta para encontrar um bom swarm, não tendo mais que programá-lo manualmente. Podendo contar, assim, com um sistema inteligente para auxiliar na descoberta dos melhores parâmetros.
6) Continuação da melhoria do código a fim de criar a primeira versão funcional da ferramenta e, assim, disponibilizá-la, finalmente, para a comunidade, recebendo melhorias dos contribuidores Open-source.

Por fim, os pesquisadores envolvidos nesse projeto agradecem ao programa unificado de bolsas da USP, por todo o financiamento da pesquisa, bem como aos biólogos que se mostraram sempre disponíveis para consulta teórica e também para contribuições criativas para a pesquisa.

# 7. Referências


1 da Silva, E. O. A., Bentes, C., Bahiense, L., & de Castro, M. C. S. (2005). Uma Abordagem Paralela Baseada em Colônia de Formigas para o Problema do Caixeiro Viajante. Cadernos Do IME-Série Informática, 18.

3 Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

4 Hiejima, T., Kawashima, S., Ke, M., & Kawahara, T. (2019). Effectiveness of Synchronization and Cooperative Behavior of Multiple Robots based on Swarm AI. 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 341–344. https://doi.org/10.1109/APCCAS47518.2019.8953108

5 Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

6 Ko, A. W. Y., & Lau, H. Y. K. (n.d.). Intelligent Robot-assisted Humanitarian Search and Rescue System.

7 Lim, S., Wang, S., Lennox, B., & Arvin, F. (2021). BeeGround - An Open-Source Simulation Platform for Large-Scale Swarm Robotics Applications. 2021 7th International Conference on Automation, Robotics and Applications (ICARA), 75–79. https://doi.org/10.1109/ICARA51699.2021.9376494

8 Majd, A., Ashraf, A., Troubitsyna, E., & Daneshtalab, M. (2018). Using Optimization, Learning, and Drone Reflexes to Maximize Safety of Swarms of Drones. 2018 IEEE Congress on Evolutionary Computation (CEC), 1–8. https://doi.org/10.1109/CEC.2018.8477920

9 Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, 25–34.

10 Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. Science, 345(6198), 795–799. https://doi.org/10.1126/science.1254295

11 Senra, R. (n.d.). Brumadinho, a história de uma tragédia que poderia ter sido evitada. Retrieved June 3, 2021, from https://www.bbc.com/portuguese/brasil-47399659

12 The Industry's Foundation for High Performance Graphics. (n.d.). Retrieved June 3, 2021, from https://www.opengl.org//

13 Yang, C., Chen, J., & Tu, X. (2007). Algorithm of Fast Marriage in Honey Bees Optimization and Convergence Analysis. 2007 IEEE International Conference on Automation and Logistics, 1794–1799. https://doi.org/10.1109/ICAL.2007.4338865

14 J. Tang, G. Liu and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 10, pp. 1627-1643, October 2021, doi: 10.1109/JAS.2021.1004129.

15 M. Janga Reddy, D. Nagesh Kumar; Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. H2Open Journal 1 January 2020; 3 (1): 135–188. doi: https://doi.org/10.2166/h2oj.2020.128

16 Diep, Q.B., Truong, T.C., Zelinka, I. (2021). Swarm Intelligence and Swarm Robotics in the Path Planning Problem. In: Piunovskiy, A., Zhang, Y. (eds) Modern Trends in Controlled Stochastic Processes:. Emergence, Complexity and Computation, vol 41. Springer, Cham. https://doi.org/10.1007/978-3-030-76928-4_16

17 Ankit Thakkar, Ritika Lohiya, Role of swarm and evolutionary algorithms for intrusion detection system: A survey, Swarm and Evolutionary Computation, Volume 53, 2020, 100631, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2019.100631.

*18 E. Soria, F. Schiano and D. Floreano, "SwarmLab: a Matlab Drone Swarm Simulator," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 8005-8011, doi: 10.1109/IROS45743.2020.9340854*

*19 The Industry's Foundation for High Performance Graphics*. (n.d.). Retrieved June 3, 2021, from https://www.opengl.org//

20 M. I. S. B. Khairat, Y. Priyadi and M. Adrian, "Usability Measurement in User Interface Design Using Heuristic Evaluation & Severity Rating (Case Study: Mobile TA Application based on MVVM)," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), 2022, pp. 0974-0979, doi: 10.1109/CCWC54503.2022.9720876.

21 GLFW®. (n.d.). , disponível em: https://www.glfw.org/, acesso em: 31/08/2022

22 DearImgui, Omar Cornut. disponível em: https://github.com/ocornut/imgui, acesso em: 31/08/2022

23 CUDA® - NVidia. (n.d.)., disponível em: https://developer.nvidia.com/cuda-downloads, acesso em: 31/08/2022