

Desenvolvendo modelo BOIDS em *JavaScript*

David Motta Guerra

Universidade de Fortaleza

Abril 2019

Resumo

Esse documento foi desenvolvido como trabalho da disciplina de Engenharia do Conhecimento do mestrado em Informática Aplicada da Universidade de Fortaleza (UNIFOR).

Inicialmente foi feita uma breve definição e apresentação das principais características de um modelo baseado em agentes - *agent-based models (ABMs)*. Em seguida foi abordado o modelo BOIDS, que é um modelo baseado em agentes e foi definido através da observação do comportamento do movimento em bando dos animais. Foi elaborado uma implementação de um algoritmo em *JavaScript* com a utilização do modelo e foram feitas algumas simulações através da alteração de parâmetros do modelo.

Por fim, foi apresentado o que pôde ser observado nos experimentos e onde podemos aplicar o modelo estudado.

1. Introdução

Será abordado neste trabalho as principais características de um modelo de sistema BOIDS. Mas, para falar desse modelo, precisamos inicialmente definir seu tipo, que se trata de um modelo baseado em agentes - *agent-based models (ABMs)*.

Segundo Sayama [1] “ABMs são indiscutivelmente a estrutura mais generalizada para modelagem e simulação de sistemas, que na verdade incluem autômatos celulares e redes dinâmicas como especiais casos”.

Esses modelos são utilizados em diversas disciplinas com o objetivo de mapear comportamentos ou situações para que posteriormente possam ser feitas previsões com base na definição do modelo. Por exemplo, observando operadores de um mercado financeiro pode-se prever com antecedência que uma determinada ação irá cair (ou subir). Ou que em situações com muita

chuva, em comunidades de baixa renda e que seja observada uma determinada situação, as chances de epidemia de uma doença específica sejam potencializadas, permitindo que o poder público tome providências sobre o tema.

Os agentes podem interagir entre si, entre o ambiente ou os dois ao mesmo tempo. Essa característica torna a definição de um ABM muito desafiadora, mas pode ser apresentada da seguinte forma: “Modelos baseados em agentes são modelos de simulação computacional que envolvem muitos Agentes” [1]. A Figura 1.1 mostra uma ilustração esquemática de um ABM.

Esses modelos devem ser utilizados apenas para um número expressivo de agentes, pois para uma quantidade pequena poderia ser resolvido por equações matemáticas, já para uma quantidade bem expressiva, a complexidade é tão elevada que deixaria inviável a resolução por fórmulas matemáticas. Portanto, a análise dos resultados obtidos com a aplicação dos ABMs é geralmente realizada usando análises estatísticas, por exemplo, executando simulações de Monte Carlo para obter distribuições de resultados medições em múltiplas condições experimentais e, em seguida, realizar teste de hipóteses para ver se houve alguma diferença significativa entre os diferentes condições experimentais.

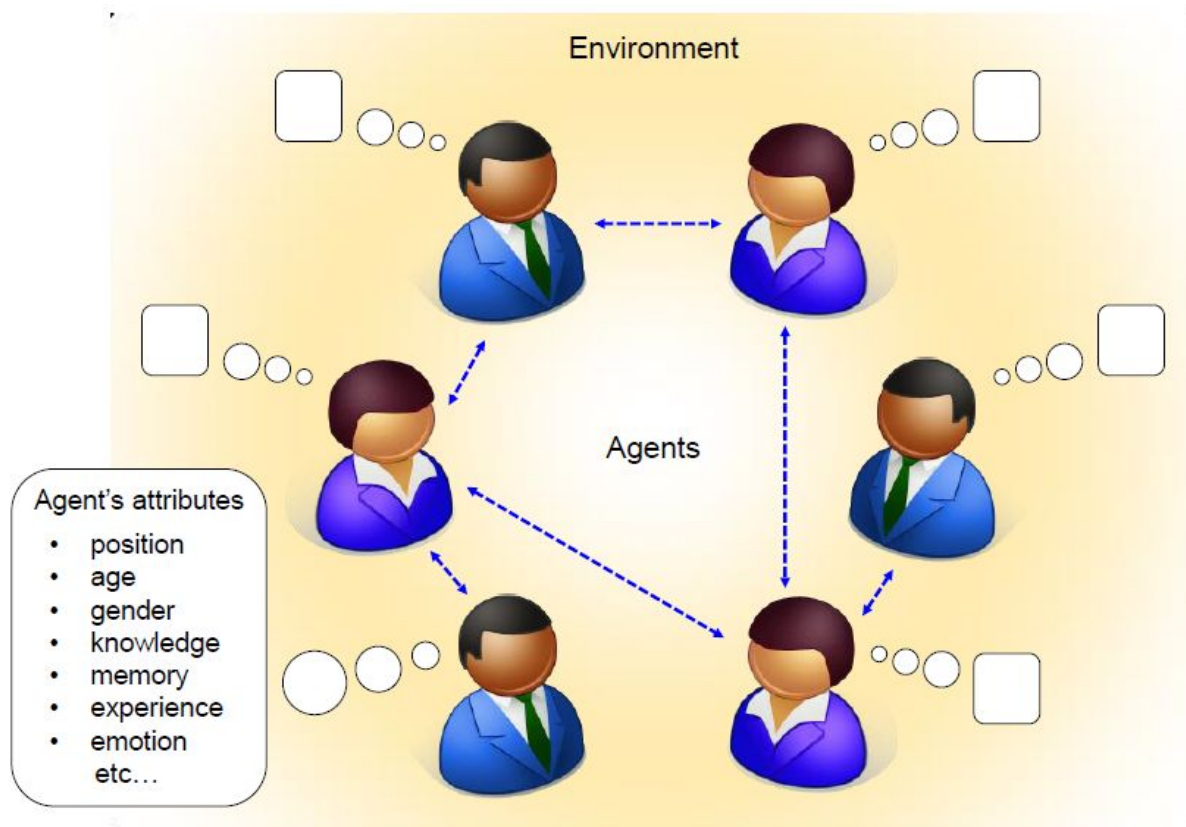


Figura 1.1: Ilustração esquemática do que é um modelo baseado em agente (ABM) [1].

A implementação de um ABM pode ser extremamente complexa, por isso é muito importante manter o código simples, organizado e de fácil manutenção. Como esses modelos são bastante flexíveis, podendo ter inúmeras propriedades e regras, existe uma tendência do desenvolvedor em adicionar cada vez mais informações ao modelo, o tornando muito complexo e, por conta dessa complexidade, também mais difícil de analisar. Para obter uma conclusão útil e confiável do modelo, é preciso encontrar um equilíbrio entre simplicidade, validade e robustez.

2. Modelo

Boids é um modelo sobre o comportamento coletivo de grupos de animais, como bandos de pássaros, cardumes de peixes e enxames de insetos. É bastante popular em pesquisas e tem sido extensivamente modelado e estudado com ABMs.

Um dos primeiros modelos computacionais desse comportamento coletivo de grupos de animais foi proposto pelo cientista da computação Craig Reynolds no final da 1980 [2]. Reynolds criou um conjunto de regras comportamentais simples para agentes em movimento em um espaço contínuo que pode reproduzir como “*an amazingly natural-looking flock behavior of birds*” (um bando incrivelmente natural do comportamento das aves). Seu modelo foi chamado de *bird-oids*, ou “boids” (pássaróide – uma mistura de pássaro com andróide [3]). Algoritmos do Boids tem sido amplamente utilizado na indústria da computação gráfica para gerar automaticamente animações de movimentos naturais de animais em bandos. Por exemplo, em cenas de filmes, como aquela do Rei Leão onde ocorre a debandada de gnus (Figura 2.2) ou nos morcegos do filme do Batman (Figura 2.1).

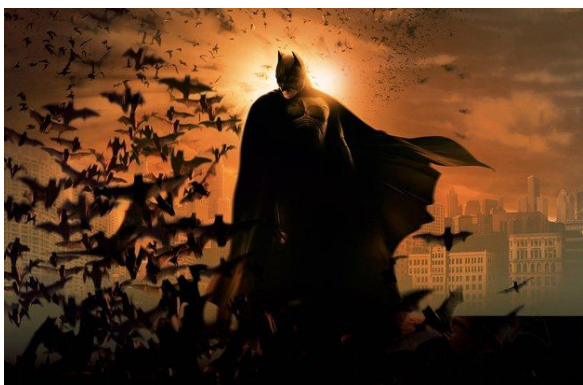


Figura 2.1: Morcegos do filme Batman



Figura 2.2: Gnus do filme Rei Leão

A dinâmica de BOIDS é gerada pelas três seguintes regras comportamentais essenciais:

- **Coesão:** agentes tendem a dirigir-se ao centro de massa dos vizinhos locais.
- **Alinhamento:** agentes tendem a dirigir para alinhar suas direções com a velocidade média de vizinhos locais.
- **Separação:** agentes tentam evitar colisões com vizinhos locais.

Agora será apresentada uma aplicação desenvolvida em *JavaScript* para experimentar o modelo BOIDS. Essa linguagem foi escolhida por ser facilmente testada e disponibilizada. Foi considerado apenas duas dimensões para facilitar a codificação.

A implementação das regras de coesão, alinhamento e separação foram feitas seguindo o artigo de Reynolds [4]. A primeira destas obtém um vetor que corresponde à média da soma das distâncias entre o pássaro (limitando os pássaros considerados em um raio informado), levando em conta que uma distância maior representa uma necessidade menor de separação. A regra de alinhamento consiste em realizar uma média das direções de deslocamento daqueles pássaros contidos no raio considerado para essa regra. Na coesão é feito uma média das posições atuais dos pássaros existentes no raio especificado para a regra, efetivamente indicando uma posição de centro de massa para onde um pássaro específico deverá ser deslocado. Em todas as regras têm-se um vetor resultante que é somado e que indica como cada pássaro deve responder ao ambiente. Entretanto, para tornar esta resposta menos artificial, os movimentos são limitados de forma que não ocorram movimentos bruscos entre um quadro e outro. Atualmente, a força máxima que esses vetores podem agir sobre os agentes estão limitadas a 0,03 (valor arbitrário) e resulta em movimento mais suaves.

Para facilitar as operações com posições (vetor) e renderização (*canvas*) foi utilizado a biblioteca “p5” [7]. Inicialmente foi criado apenas um arquivo HTML com a importação da biblioteca e com o elemento onde será renderizada a imagem com os pássaros.

Código 2.1

```
<script src="p5.min.js" type="text/javascript"></script>
<body style="margin: 0">
  <div id="boids">
    </div>
</body>
```

Dessa parte em diante o conteúdo será apenas em *JavaScript*. Foram criadas duas classes: “Passaro” e “Bando”. A primeira representará cada elemento no modelo e a segunda a junção desses pássaros.

Ao inicializar a classe “Passaro”, ela é posicionada em “x, y” e com uma velocidade aleatória.

Código 2.2

```
class Passaro {  
  
    // Inicializa os parâmetros  
    constructor(x, y) {  
        this.x = x;  
        this.y = y;  
        this.aceleracao = createVector(0, 0);  
        this.velocidade = createVector(random(-1, 1), random(-1, 1));  
        this.posicao = createVector(x, y);  
    }  
}
```

Foi adicionado método “movimentar” que irá fazer toda a verificação das regras principais do modelo BOIDS: separação, alinhamento e coesão. Após a aplicação das regras, será verificado se o pássaro ficou fora dos limites da imagem (caso tenha ficado ele terá sua posição alterada para a próxima extremidade) e renderizado na tela.

Código 2.3

```
    // Desloca o pássaro conforme análise de separação, alinhamento e  
    coesão  
    movimentar(passaros) {  
        this.analisarVizinhanca(passaros);  
        this.atualizarPosicao();  
        this.verificarLimites();  
        this.renderizar();  
    }  
  
    // Verifica a influência dos pássaros vizinhos e adiciona na  
    aceleração
```

```

    analisarVizinhanca(passaros) {
        this.aceleracao.add(this.separacao(passaros));
        this.aceleracao.add(this.alinhamento(passaros));
        this.aceleracao.add(this.coesao(passaros));
    }

```

Foi criada a classe “Bando” para controlar todos os pássaros. É através dela que a cada atualização de um quadro será chamado o método de movimento de todos os pássaros do modelo.

Código 2.4

```

class Bando {

    passaros = [];

    movimentar() {
        this.passaros.forEach(passaro => {
            passaro.movimentar(this.passaros);
        });
    }

    adicionarPassaro(b) {
        this.passaros.push(b);
    }

}

```

E, por fim, foi criado um trecho de código para instanciar o bando com um determinado número de pássaros. Foi definido também uma velocidade máxima que cada pássaro pode possuir e a força máxima que os seus vizinhos podem exercer sobre ele.

Código 2.5

```

function setup() {
    [1,2,3,4,5,6,7,8].forEach(b => passarosImg.push(loadImage('img/' +
    b + '.png')));
}

```

```
let canvas = createCanvas(width, heightCanvas);
canvas.parent("boids"); // Adiciona a imagem no DIV
init();
}

function init() {
  iteracoes = 0;
  inicio = (new Date()).getTime();
  let qtdPassaros = 150;
  bando = new Bando();
  for (let i = 0; i < qtdPassaros; i++) {
    let passaro = new Passaro(Math.random() * width, 0);
    passaro.velocidadeMaxima = 2;
    passaro.forcaMaxima = 0.03;
    passaro.raioSeparacao = 25;
    passaro.raioAlinhamento = 50;
    passaro.raioCoesao = 50;
    bando.adicionarPassaro(passaro);
  }
}
```

Para não deixar o conteúdo deste trabalho muito extenso, os códigos referentes aos métodos de coesão, alinhamento e separação não foram incluídos, mas o conteúdo completo pode ser acessado através do endereço <https://davidmguerra.github.io/mestrado/> (na opção Engenharia do Conhecimento / BOIDS).

3. Resultados

Foram feitos alguns experimentos através da modificação dos parâmetros de tamanho da população, velocidade e raios de separação, alinhamento e coesão. A cada execução foi registrado a situação após cerca de 500 mil iterações. Inicialmente foi utilizado como parâmetro um determinado tempo, mas foi observado que a medida que a população era aumentada o sistema rodava de forma mais lenta (pois a complexidade do algoritmo é $O(n^2)$). Então esse parâmetro não se mostrou muito eficaz.

A primeira execução foi realizada com uma população de 150 pássaros, raios de coesão, alinhamento e separação de 50, 50, 25 e velocidade máxima de 2 (significa que o pássaro vai se

deslocar no máximo 2 pixels em qualquer direção a cada iteração). O resultado pode ser verificado na Figura 3.1.

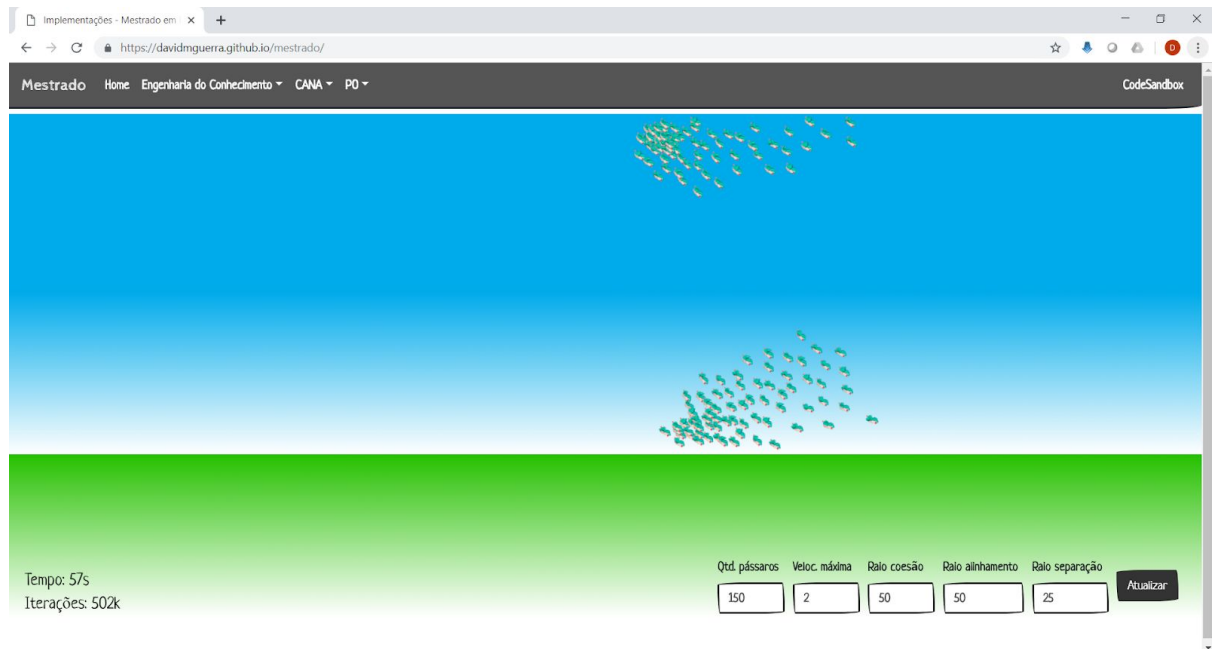


Figura 3.1: 150 pássaros, velocidade máxima de 2 e raios 50, 50 e 25.

No segundo experimento foi quadruplicado a quantidade de pássaros, passando para 600. Foi observado que com o número elevado de pássaros a força que uns exercem sobre os outros aumenta muito e gera mais agrupamentos distintos que quando executado com um número menor, conforme observado na Figura 3.2.

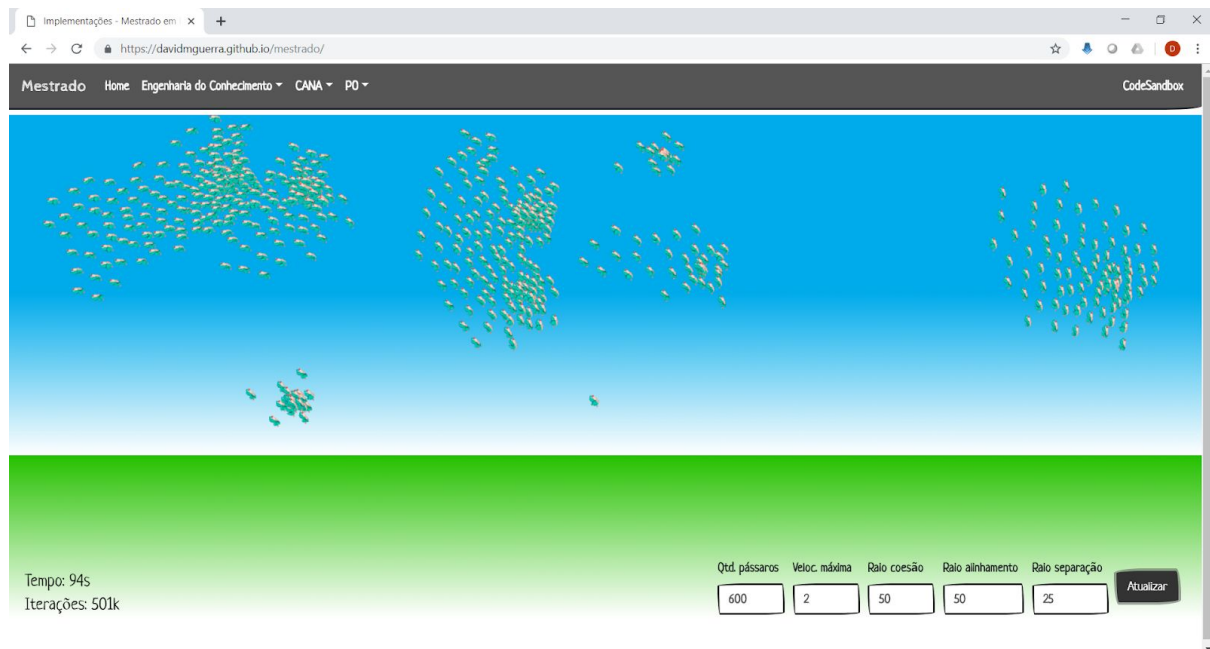


Figura 3.2: 600 pássaros, velocidade máxima de 2 e raios 50, 50 e 25.

Em seguida os valores dos raios de coesão, alinhamento e separação foram dobrados. Com essa alteração foi observado que aumentou o agrupamento, conforme mostrado na Figura 3.3.

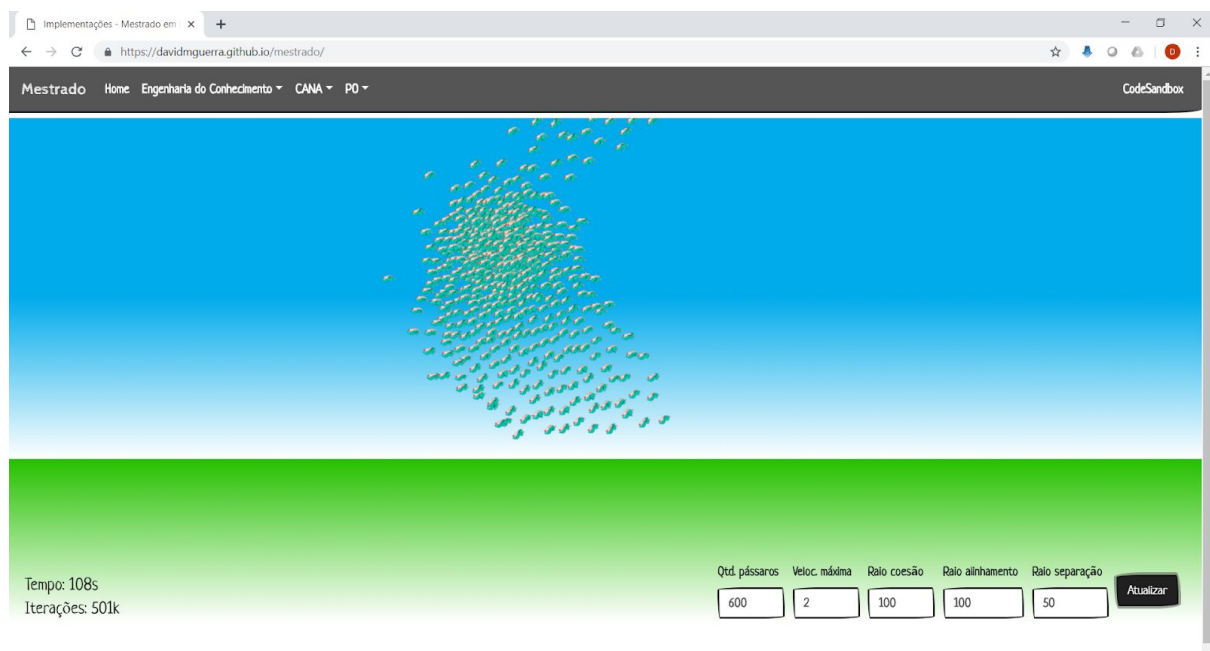


Figura 3.3: 600 pássaros, velocidade máxima de 2 e raios 100, 100 e 50.

Com os mesmos parâmetros de quantidade de pássaros e raios do experimento anterior, foi quadruplicado o parâmetro de velocidade máxima. Foi observado que a agregação do bando foi reduzida e a formação ficou mais esparsa.

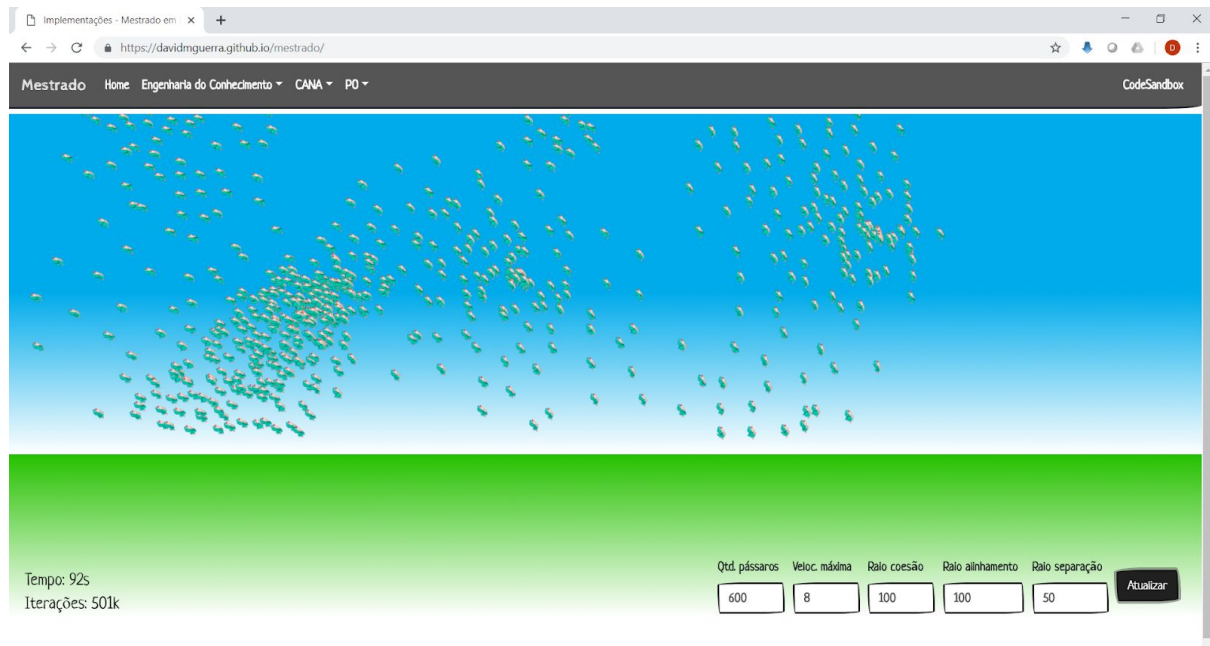


Figura 3.3: 600 pássaros, velocidade máxima de 8 e raios 100, 100 e 50.

4. Discussões

Foi verificado que quando se aumenta muito a quantidade de agentes do modelo ocorre uma tendência de criação de grupos distintos, elevando a segregação. Mas, por outro lado, pode-se deixar o grupo menos esparsa aumentando os raios de coesão, alinhamento e separação, que são os limites que os agentes podem ser afetados por outros agentes.

Também foi verificado que quando a velocidade máxima é elevada os agentes ficam mais esparsos.

Foi observado também que muito da segregação que ocorre é devido ao método que limita os agentes no “céu” e, quando ele vai em direção a uma área fora das delimitações, ele é reinserido na outra extremidade.

5. Conclusões

O modelo boids é um exemplo de um modelo baseado em indivíduos, uma classe de simulação usada para capturar o comportamento global de um grande número de agentes autônomos interagentes. Modelos baseados em indivíduos estão sendo usados em biologia, ecologia, economia e outras áreas de estudo.

Não foram utilizados todos os conceitos apresentados por Craig Reynolds [4], como a utilização de obstáculos, mas foram experimentados as principais regras (coesão, alinhamento e separação) e aplicado os conceitos de ABMs.

Por o algoritmo ter complexidade $O(n^2)$ ele pode se tornar inviável para uma quantidade muito grande de agentes. No entanto, é possível reduzir esse custo para quase $O(n)$ pelo uso de uma estrutura de dados espaciais adequada que permite que os BOIDS sejam ordenados por sua localização. Encontrar os vizinhos próximos de um BOID requer que se examine apenas a porção do lote que está dentro da vizinhança geral. Usando essas acelerações algorítmicas e hardware rápido e moderno, grandes bandos podem ser simulados em tempo real, permitindo aplicações interativas [4].

Referências

[1] Hiroki Sayama. Introduction of the Modeling and Analysis of Complex Systems. Open SUNY Textbooks, 2015.

[2] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," ACM SIGGRAPH Computer Graphics, vol. 21, no. 4, pp. 25–34, 1987.

[3] gpolo/birdflocking. Disponível: <https://github.com/gpolo/birdflocking>

[4] Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model). Disponível: <https://www.red3d.com/cwr/boids/>

[5] Flocking \ Examples \ Processing.org. Disponível: <https://processing.org/examples/flocking.html>

[6] Boids Pseudocode. Disponível: <http://www.kfish.org/boids/pseudocode.html>

[7] p5.js | home. Disponível: <https://p5js.org/>