

从零开始的蚁群算法

the_Afish

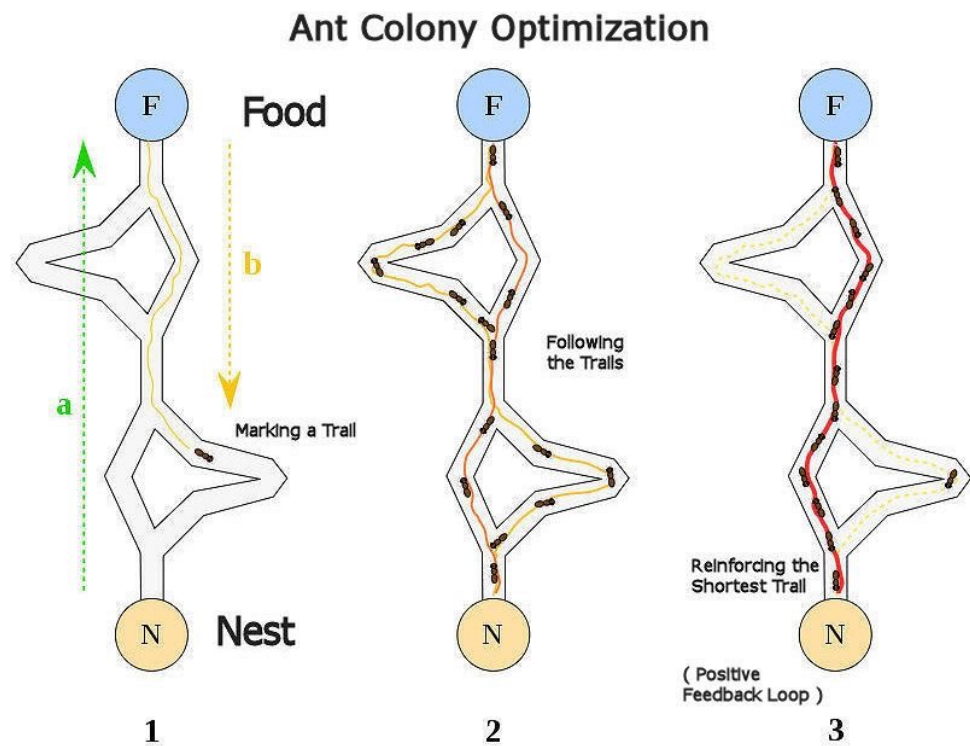
2021/10/28

什么是蚁群算法？

Ant Colony Optimization (ACO)

Rules?

1. 蚂蚁会释放两种信息素
2. 觅食蚂蚁会追寻“食物”信息素，并释放“回家”信息素
3. 归家蚂蚁会追寻“回家”信息素，并释放“食物”信息素
4. 信息素会随时间逐渐挥发



http://en.wikipedia.org/wiki/Ant_colony_optimization

什么是蚁群算法？

Ant Colony Optimization (ACO)

IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol.26, No.1, 1996, pp.1-13 1

The Ant System: Optimization by a colony of cooperating agents

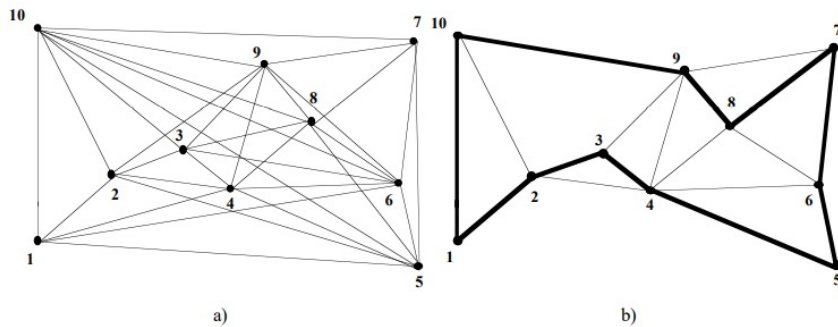
Marco Dorigo^{*,^}, Member, IEEE, Vittorio Maniezzo^{%,^}, and Alberto Colomi[#]

* IRIDIA, Université Libre de Bruxelles, Avenue Franklin Roosevelt 50, CP 194/6, 1050 Brussels, Belgium.
Email: mdorigo@ulb.ac.be URL: <http://iridia.ulb.ac.be/dorigo/dorigo.html>

[^] Politecnico di Milano Artificial Intelligence and Robotics Project, Dipartimento di Elettronica e Informazione,
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy.

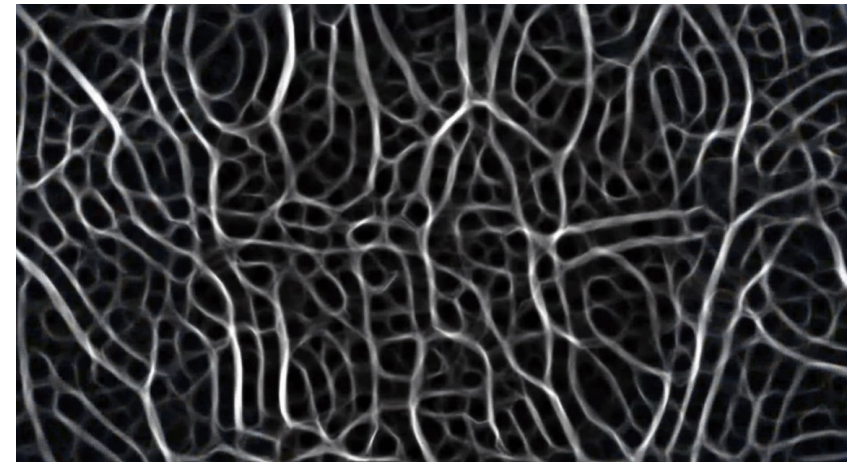
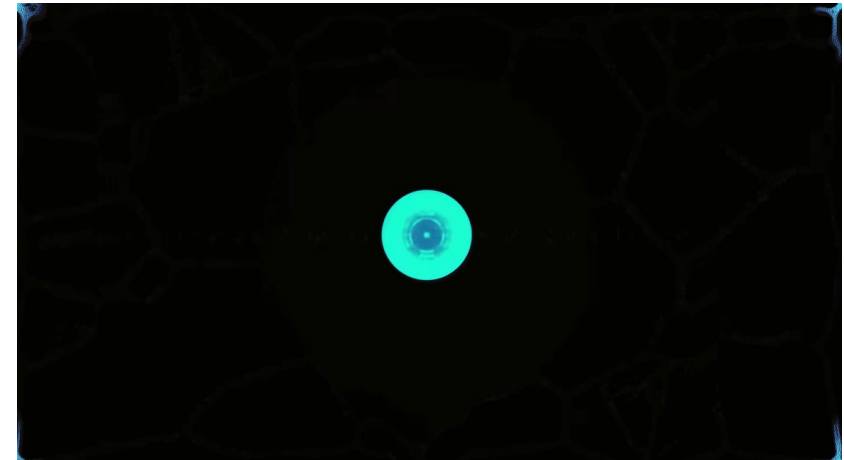
[%] Dipartimento di Scienze dell'Informazione, Università di Bologna, Via Sacchi 3, 47023 Cesena, Italy. Email:
maniezzo@csr.unibo.it

[#] Centro di Teoria dei Sistemi del CNR, Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy. Email: colomi@elet.polimi.it



DOI: 10.1109/3477.484436

利用类似策略可模拟黏菌



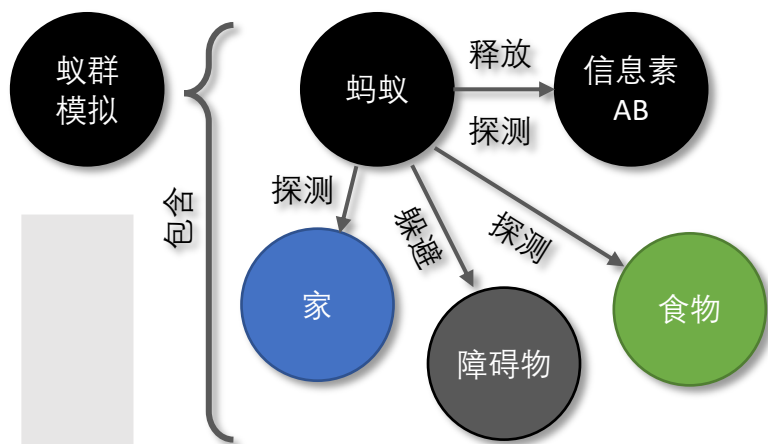
<https://www.youtube.com/watch?v=X-iSQQgOd1A&t=513s>

DOI: 10.1162/artl.2010.16.2.16202

尝试

不参照其他文献以及代码实现ACO？

组成分析：



```
class AntColony: ...  
class Ants: ...  
class Pheromone: ...  
class Food: ...  
class Obstacle: ...
```

最初计划：

蚁窝、蚂蚁、食物、障碍：

`pos = ti.Vector.field()`

`radius = r`

信息素：

`grid = ti.field()`

蚂蚁的能力：

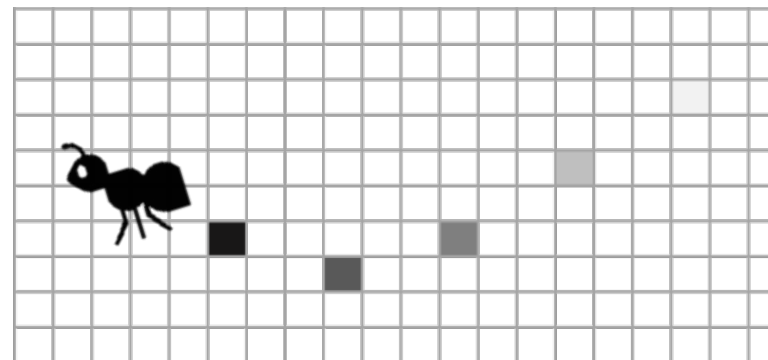
`random_walk()`

`change_direction()`

`detect_pheromone(type)`

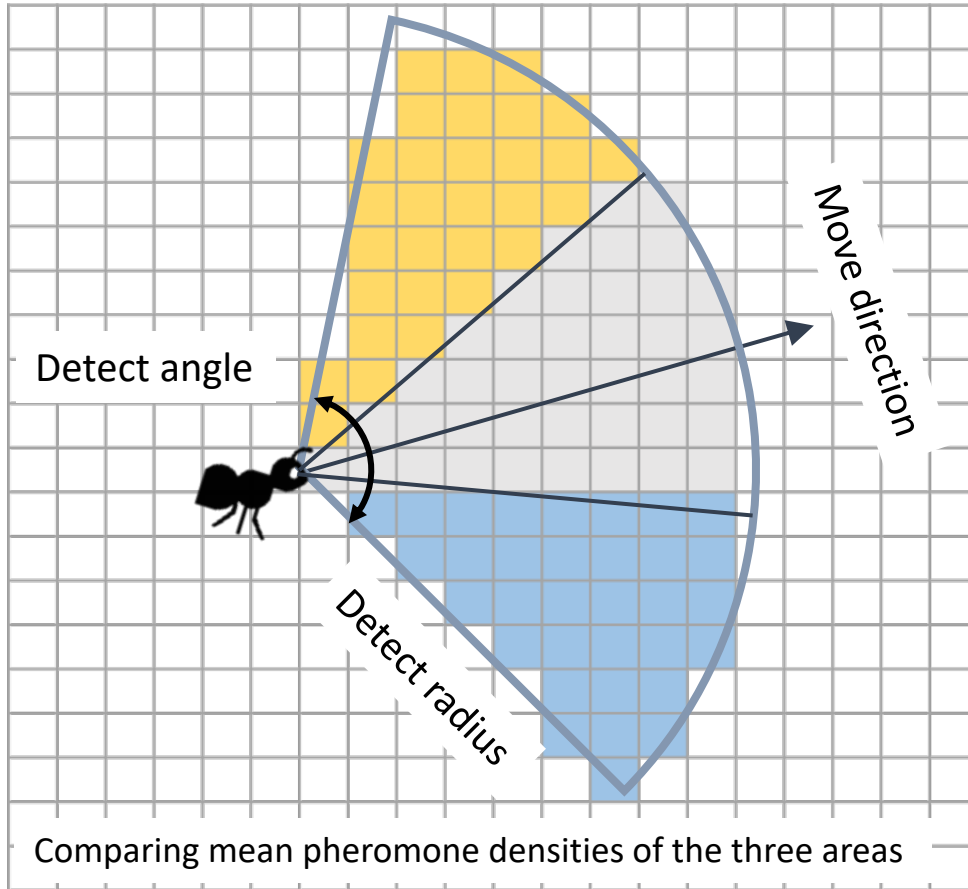
`detect(food/obstacle/home)`

`release_pheromone()`

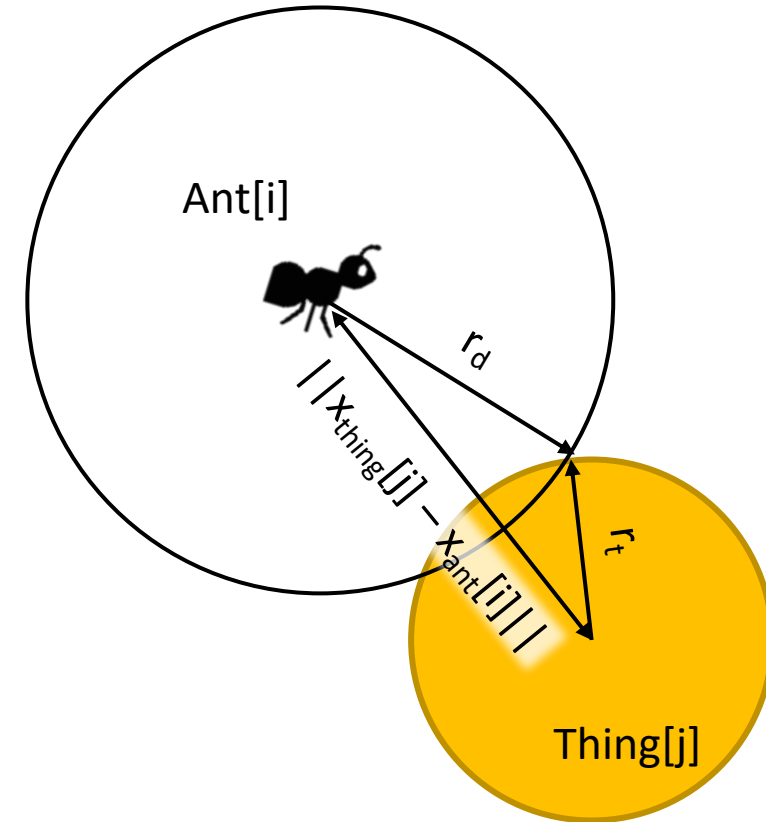


How to Detect?

For Pheromones



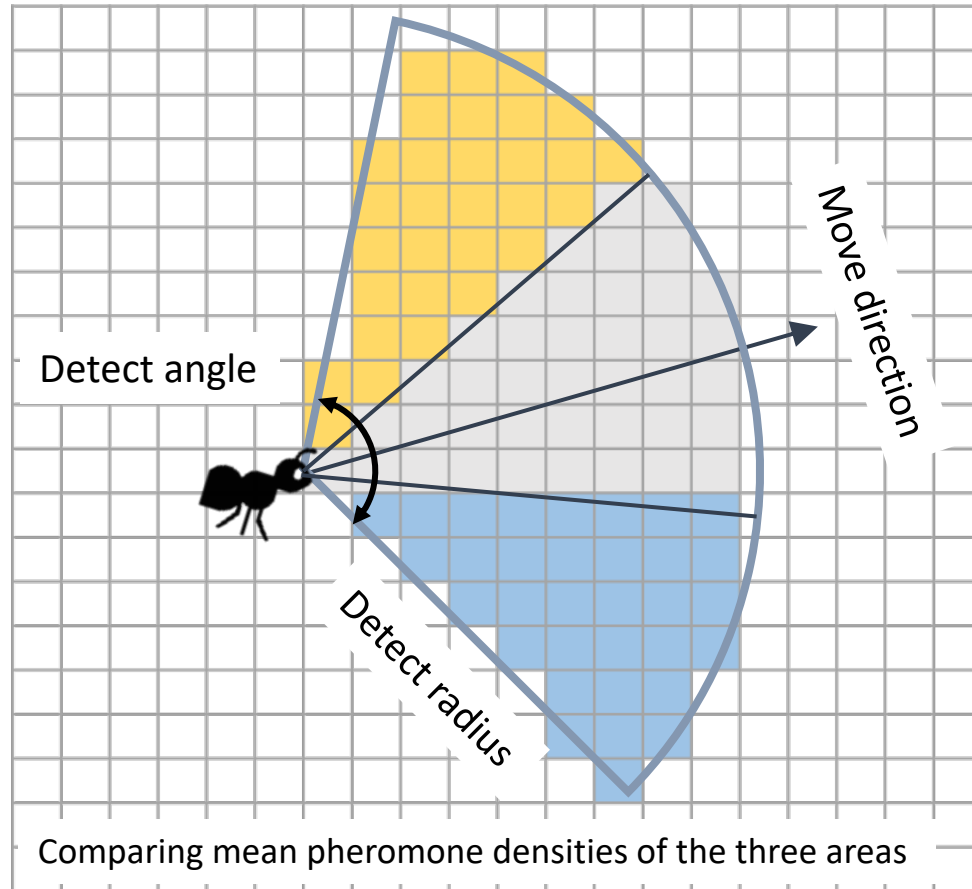
For Foods, Home and Obstacles



$$r_d + r_t > \|x_{thing[j]} - x_{ant[i]}\| ? \text{ detected : undetected}$$

How to Detect?

For Pheromones



Taichi 极其容易提升检测效率：

- 1、Block-major 存储信息素
- 2、稀疏计算减少对空格点的遍历

相较于C++，代码修改贼拉快

```
block1 = ti.root.pointer(ti.ij, 8)
block2 = block1.pointer(ti.ij, 4)
block3 = block2.pointer(ti.ij, 4)
block3.dense(ti.ij, 4).place(pheromone)
```

Wait...

Food

Obstacle

Pheromone

...

Detectables!

```
class Pheromone: ...  
class Food: ...  
class Obstacle: ...
```

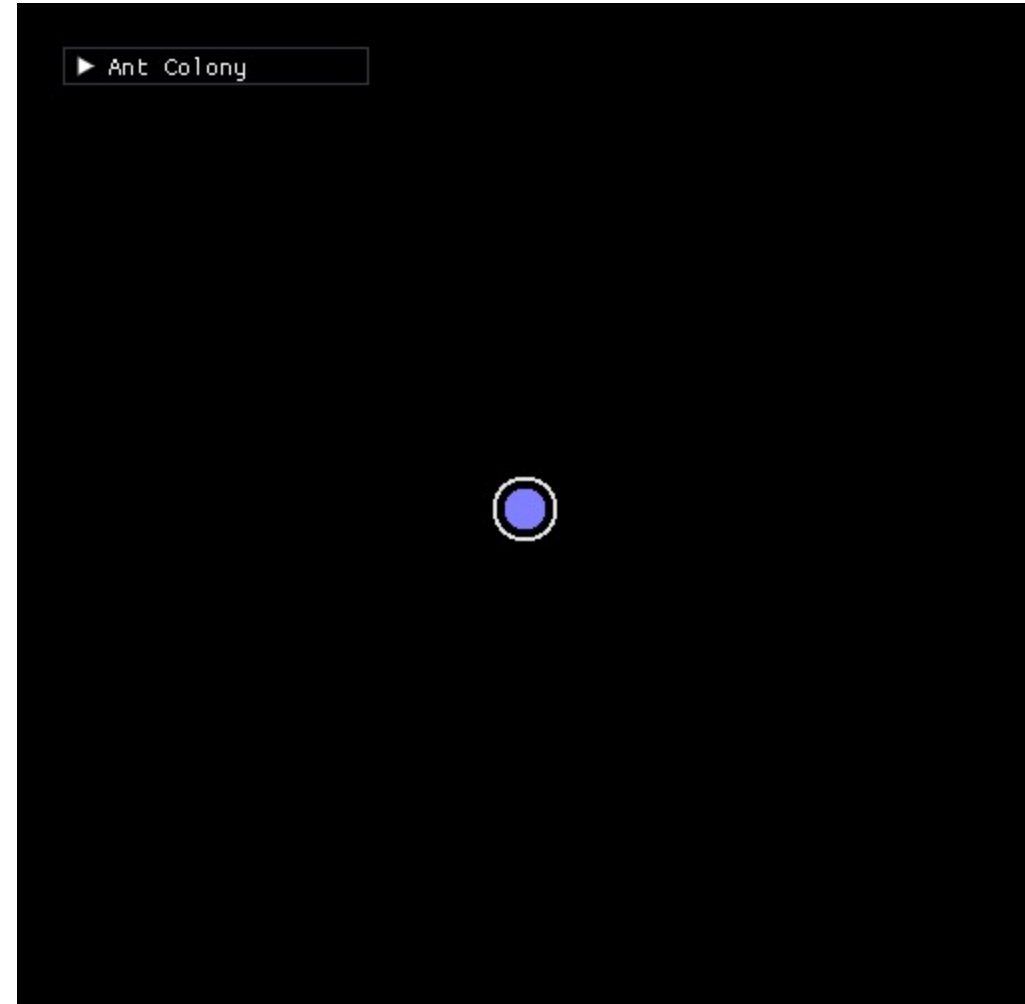


```
class Detectables: ...
```

```
detect_pheromone(typ)  
detect_circle(typ)
```



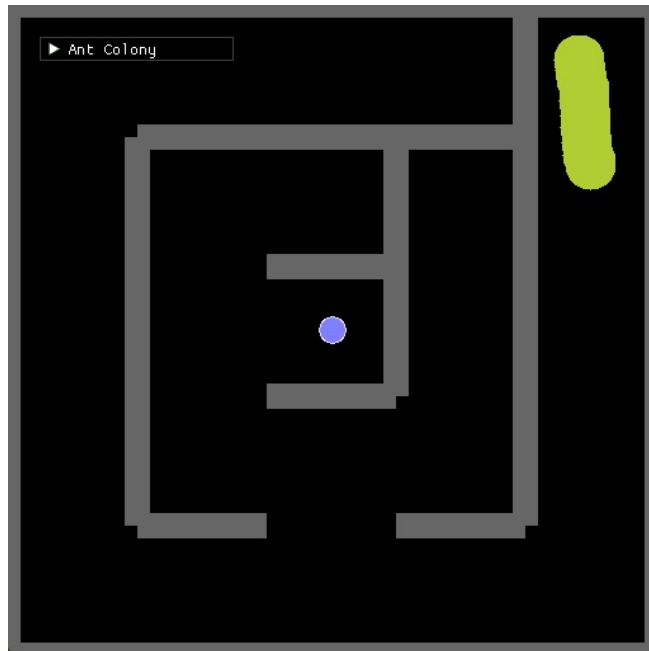
```
detect_things(typ)
```



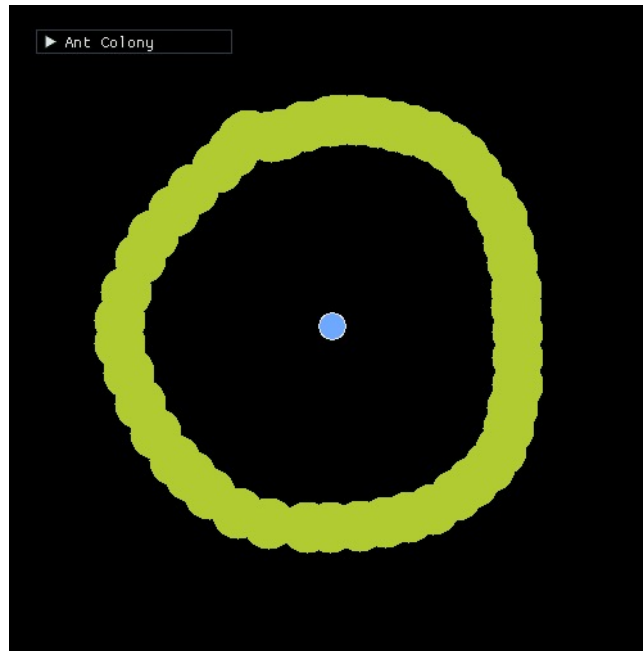
Gallery

with GGUI, Superfast

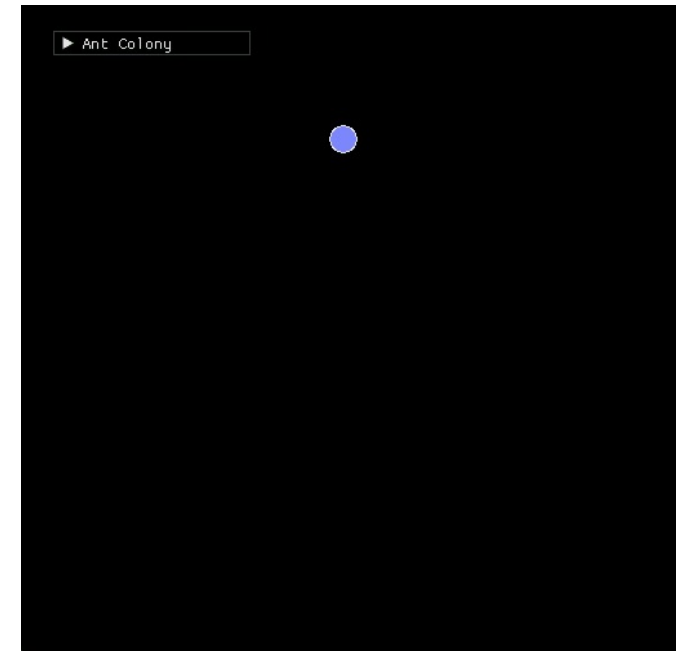
Simple Puzzle



Wheel Shape

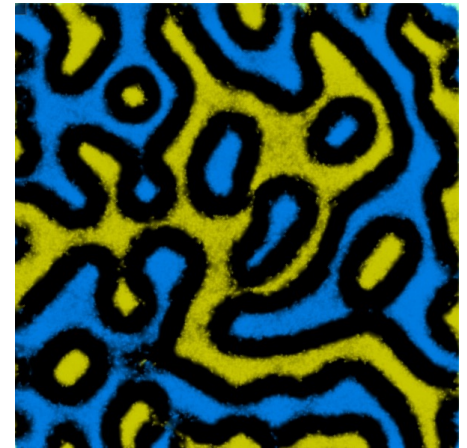
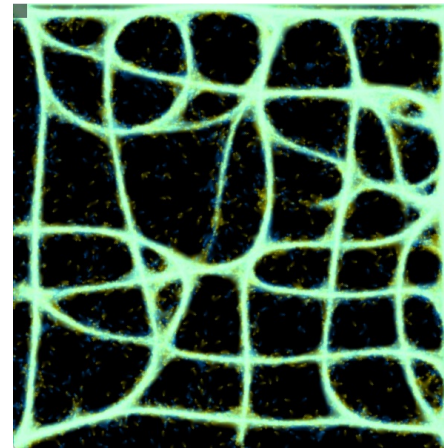
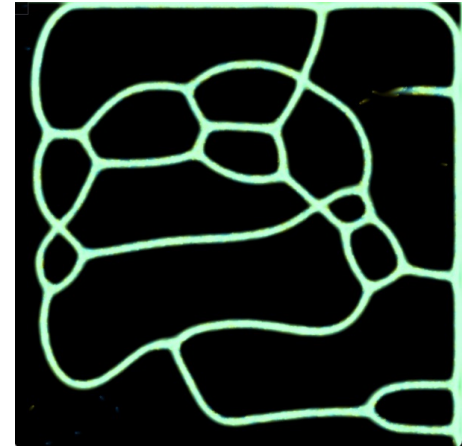
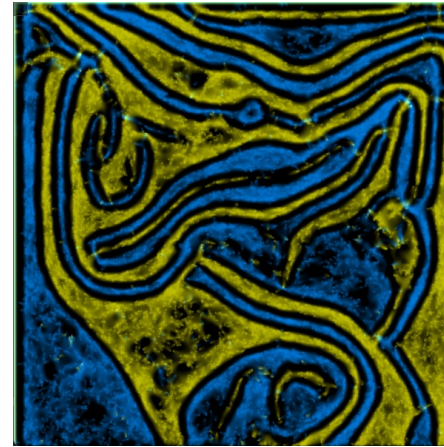
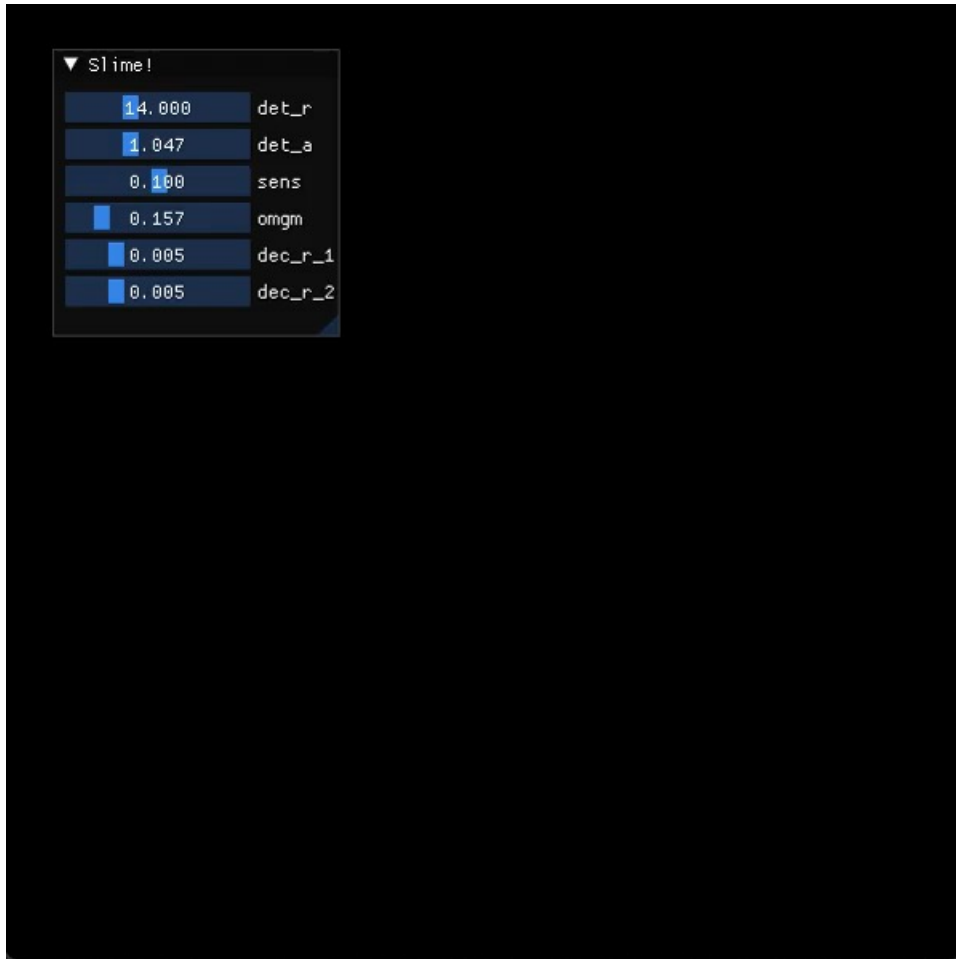


Food Suddenly Appears

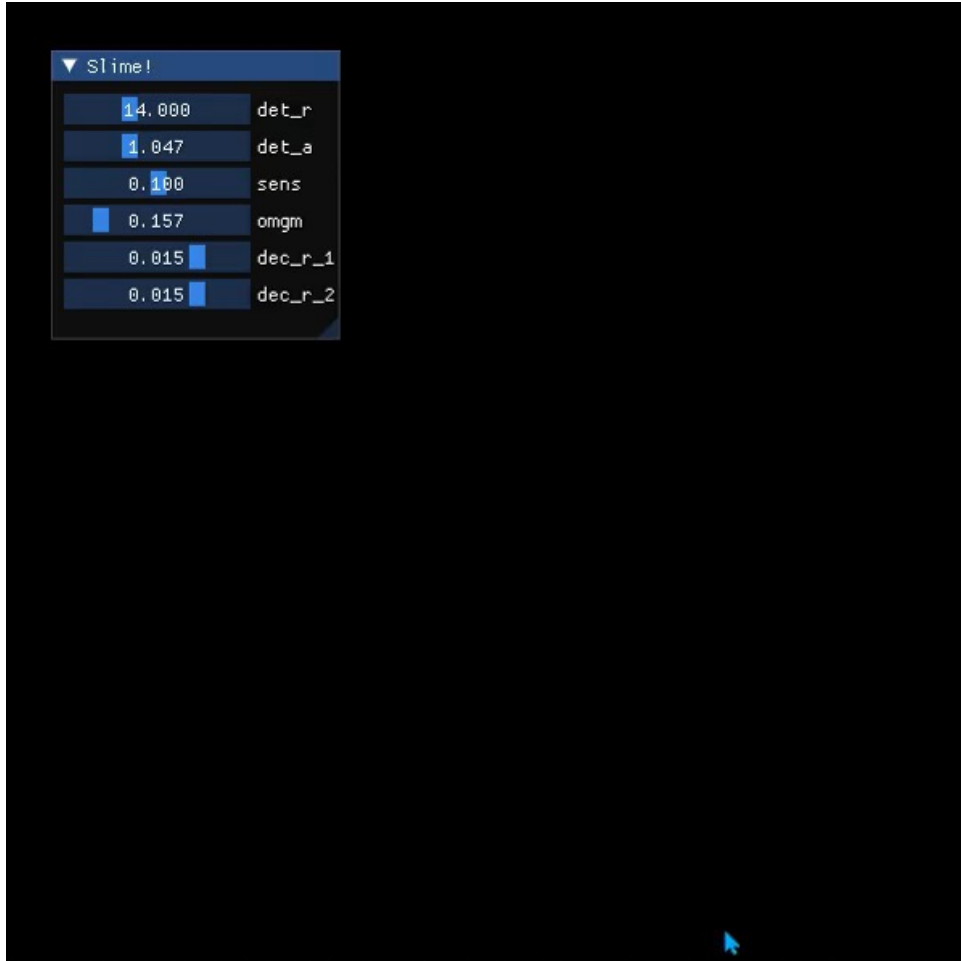


Slime Simulation! (30000 ants)

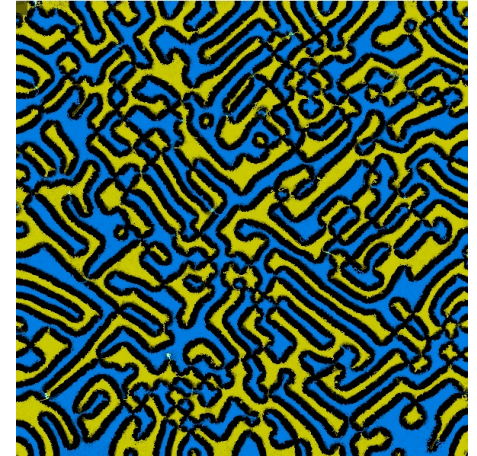
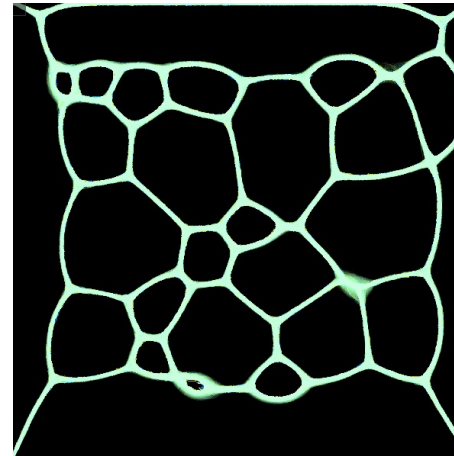
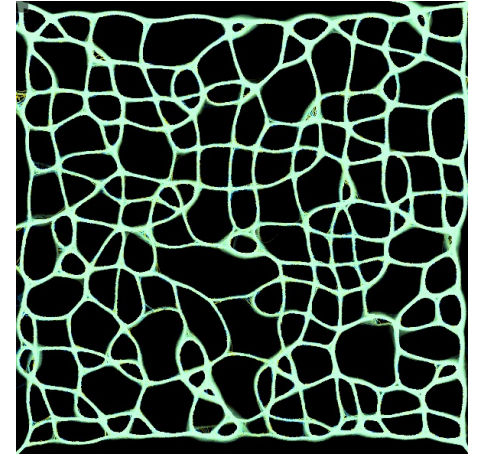
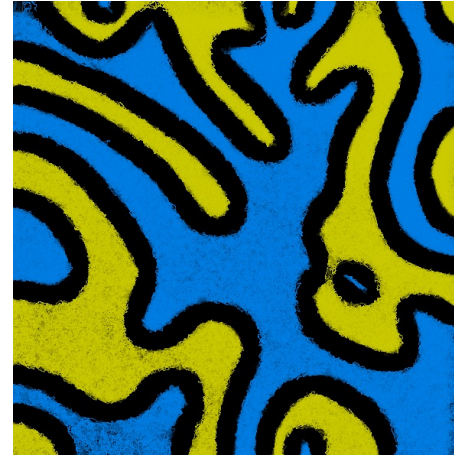
如果去掉食物和家会发生什么 ? (with blur)



Slime Simulation!



如果去掉食物和家会发生什么？(w/o blur)



仍有很多问题

1. 蚂蚁与障碍物之间的交互
2. 寻路效率
3. 代码结构优化
4. 画笔效率
5. 黏菌模拟不够好看
6.

