

## ЗАДАНИЕ 6

### Модель муравьиной колонии

---

**Построение модели** • Рассмотрим реализацию модели поиска пищи колонией муравьев, в которой муравьи помечают пройденный путь двумя видами феромона, один используется в процессе поиска пищи (предполагается, что источников пищи может быть несколько), другой — при возвращении в муравейник. Наша модель не будет содержать препятствий, поэтому смоделировать эксперимент с двумя мостами в рамках этой модели у нас не получится, однако мы рассмотрим эту задачу в упражнениях.

**A** Создаем новую модель. Устанавливаем размеры `max-pxcor` и `max-ycor` равными 50, размер патча — 5 пикселям.

**B** Создаем глобальные переменные `nest` и `food`, в которых с целью удобства программирования будем хранить патч, представляющий муравейник, и набор патчей, представляющих источники пищи.

**C** Создаем новый вид черепах — муравьев (`ants` и `ant`). Приписываем муравьям два атрибута:

- `ph` — текущий уровень феромонной активности муравья (чем дольше ходит муравей в поисках гнезда или пищи, тем меньше феромона он откладывает на своем пути);
- `state` — состояние муравья ("`forward`", если муравей ищет пищу, "`backward`", если возвращается назад).

**D** Приписываем патчам три атрибута:

- **p-type** — тип патча, принимает одно из трех значений: **"free"**, **"nest"** или **"food"**;
- **ph-n** — уровень феромона первого типа, которым муравьи помечают путь *из* муравейника;
- **ph-f** — уровень феромона второго типа, которым муравьи помечают обратный путь *в* муравейник.

Максимальное значение уровня феромона каждого типа будет равно 1, минимальное — 0.

**E** Создаем процедуру **setup**, в которой очищаем мир, сбрасываем таймер. К интерфейсу модели добавляем кнопку **setup**. Переходим к написанию кода процедуры **setup**.

**F** Устанавливаем тип **p-type** всех патчей равным значению **"free"**.

**G** Определяем патчи, представляющие муравейник (центральный патч модели) и источники пищи. Последние будем выбирать случайным образом на расстоянии, не меньшем  $0.8 * \text{max-pxcor}$  от муравейника. Число источников пищи **food-sources** будем определять с помощью соответствующего слайдера.

```
1 set nest patch 0 0
2 set food n-of food-sources patches with
3   [distance nest > (max-pxcor * 0.8)]
```

**H** Просим патч-муравейник установить тип равным значению **"nest"** и создать с помощью команды **sprout** в его центре черепаху размера 4, голубого (**sky**) цвета, имеющую форму **"circle 3"**<sup>1</sup>.

**I** Аналогичным образом просим патчи, представляющие источники пищи, установить тип равным значению **"food"** и создать в своем центре черепаху размера 4, красного цвета, имеющую форму **"circle 3"**.

**J** Просим патчи изменить свой цвет с помощью команды **color-patch**. Создаем соответствующую процедуру **color-patch**, в которой определяем цвет патча, согласно усредненному уровню его феромонов **ph-n** и **ph-f**, белый цвет будет соответствовать минимуму феромона, зеленый — максимуму:

```
1 to color-patch
2   let p (ph-n + ph-f) / 2
3   let c 255 * (1 - p)
```

<sup>1</sup> Такую форму, состоящую из двух кругов (рис. 6.1), надо создать самостоятельно в редакторе формул. Большой круг должен быть черного цвета, меньший — серого (переменного).

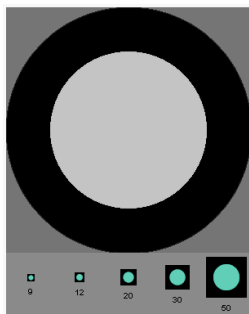


РИС. 6.1 Форма **"circle 3"**

```
4 set pcolor rgb c 255 c
5 end
```

**К** Добавляем к интерфейсу слайдер **ants-number** с диапазоном от 0 до 1000, представляющий размер колонии муравьев. В процедуре **setup** создаем соответствующее число муравьев, располагая их в центре модели (т.е. в муравейнике). Устанавливаем размер муравьев равным 2, форму — "bug".

**L** Проверяем работу кнопки **setup** (рис. 6.2, все муравьи пока скрыты под голубым кругом в центре модели).

**М** Добавляем к интерфейсу кнопку **go** и слайдер **ant-vel** с диапазоном от 0 до 10, представляющий скорость муравьев. Пишем код процедуры **go**, в котором просим всех муравьев передвинуться вперед на расстояние **ant-vel**, обновляем таймер. Т.к. все муравьи изначально сосредоточены в одной точке, а направления движения у них случайные, то движение всей колонии будет иметь ярко выраженный волновой вид (рис. 6.3).

**N** Переходим к написанию кода работы с феромоном. В процедуре **go** перед командой движения муравья (см. предыдущий пункт) проверяем, находится ли муравей в гнезде или в источнике пищи, если находится, то устанавливаем уровень его феромона **ph** равным 0.1:

```
1 if p-type = "nest" or p-type = "food" [
2   set ph 0.1
3 ]
```

После этого вызываем процедуру **rotate-ant**. После команды движения муравья вызываем процедуру **put-pheromone**.

**O** Дальше, в процедуре **go**, выполняем диффузию обоих типов феромона с коэффициентом диффузии 0.01:

```
1 diffuse ph-n 0.01
2 diffuse ph-f 0.01
```

**P** После этого просим все патчи смоделировать процесс испарения феромона (обоих типов) умножением его уровня (**ph-n** или **ph-f**) на коэффициент испарения 0.99. При этом, если новый уровень оказался меньше порогового значения 0.001,

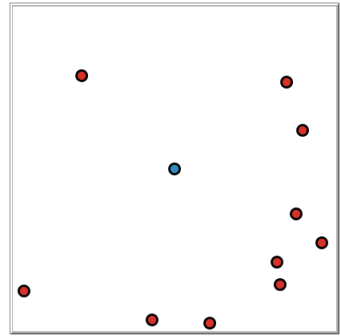


РИС. 6.2 Начальная конфигурация модели

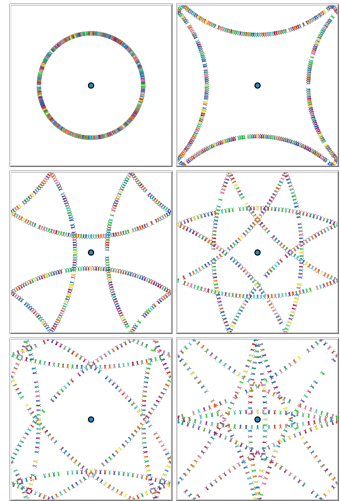


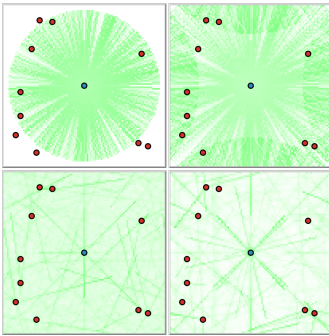
РИС. 6.3 Прямолинейное движение муравьев

<sup>2</sup> Таким образом мы моделируем разрешающую способность муравьев распознавания ими количества феромона.

сбрасываем этот уровень до нуля<sup>2</sup>. Наконец, просим все патчи выполнить команду изменения цвета `color-patch`.

**Q** Пишем код процедуры `put-pheromone`. Если текущее состояние муравья — поиск пищи (т.е. значение `state` равно `"forward"`), то прибавляем к уровню феромона `ph-n` текущего патча значение `ph`. При этом если новое значение `ph-n` превышает максимальное значение 1, то сбрасываем его до этого максимума:

```
1 if state = "forward" [
2   set ph-n ph-n + ph
3   if ph-n > 1 [set ph-n 1]
4 ]
```



**РИС. 6.4** Изменение феромона в модели (муравьи скрыты для наглядности)

Аналогичные действия над феромоном `ph-f` выполняем, если текущее состояние — возвращение в муравейник. Последней командой данной процедуры уменьшаем феромонную активность муравья, умножая его значение `ph` на коэффициент 0.99.

**R** Проверяем работу модели (желательно с максимальным размером муравьиной колонии). При этом должно быть видно (рис. 6.4), что муравьи, проходящие через муравейник или источник пищи, начинают оставлять за собой феромонный след.

**S** Наконец, переходим к написанию кода процедуры `rotate-ant`, по сути, центральной в данной модели, т.к. именно в ней реализуется логика модели, основанная на принципах стигмергии. Целью данной процедуры является изменение направления движения муравья в зависимости от его состояния и состояния окружающей его среды (т.е. уровня феромона в некоторой его локальной окрестности). Сначала проверяем, находится ли муравей в муравейнике или в источнике пищи. В первом случае меняем его состояние на `"forward"`, меняем цвет муравья на красный, меняем направление движения на обратное (команда `rt 180`) и выходим из процедуры (команда `stop`). Во втором случае делаем аналогичные действия, но меняем цвет на синий и делаем новое состояние муравья равным `"backward"`.

**T** Введем в модель параметр `vision` (добавив к интерфейсу соответствующий слайдер с диапазоном от 2 до 10), представляющий радиус поля зре-

ния муравья. Если цель (муравейник или источник пищи, в зависимости от текущего состояния) находится в поле зрения муравья, то он меняет свое направление на направление на цель (и выходит из процедуры):

```

1 if state = "forward" [
2   let f min-one-of food [distance myself]
3   if distance f < vision [
4     set heading towards f
5     stop
6   ]
7 ]
8 if state = "backward" [
9   if distance nest < vision [
10    set heading towards nest
11    stop
12   ]
13 ]

```

В первом условном операторе целью является источник пищи. Сначала мы ищем ближайший источник пищи и, если найденный источник находится в поле зрения, меняем направление на этот источник и завершаем процедуру. Второй условный оператор обрабатывает ситуацию, когда целью является муравейник.

**У** Если в области видимости цели нет, то движение муравья будет диктоваться уровнем феромона вокруг него. Для этого добавим к интерфейсу модели слайдер `angle` с диапазоном от 90 до 180, который будет представлять угол обзора муравья. Муравей выбирает патч в *конусе* видимости с радиусом от `vision / 2` до `vision` и углом `angle`, имеющий максимальный уровень феромона, соответствующего текущей цели муравья. После этого муравей поворачивается в направлении к найденному патчу.

```

1 let p patches in-cone vision angle
2   with [distance myself > vision / 2]
3 if state = "forward" [
4   set heading towards max-one-of p [ph-f]
5 ]
6 if state = "backward" [
7   set heading towards max-one-of p [ph-n]
8 ]

```

**У** На этом построение нашей модели завершено. Ее окончательный интерфейс показан на рис. 6.5. Проверяем работу модели. Пример коллективного

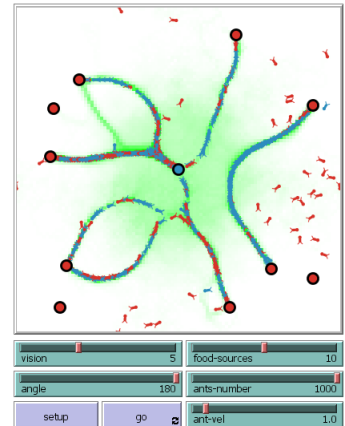


РИС. 6.5 Окончательный интерфейс модели

поведения колонии в данной модели приведен на рис. 6.6.

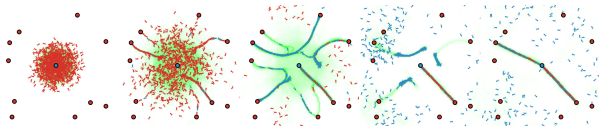


РИС. 6.6 Пример работы модели

## УПРАЖНЕНИЯ

**1** Построенная нами модель оперирует большим числом параметров, встроенных прямо в модель. Например, коэффициент испарения феромона, коэффициент диффузии, порог распознавания феромона и т.п. Сделайте для всех этих параметров соответствующие слайдеры или элементы ввода. Поэкспериментируйте с моделью, изменяя значения ее параметров.

**2** В нашей модели муравей, попадая в муравейник или источник пищи, поворачивает в противоположную сторону. Измените модель, чтобы муравей в данной ситуации делал бы поворот на случайный угол. В такой вариации намного чаще и быстрее вся колония будет приходить к консенсусу, т.е. выбирать один (обычно ближайший к гнезду) источник пищи (рис. 6.7).

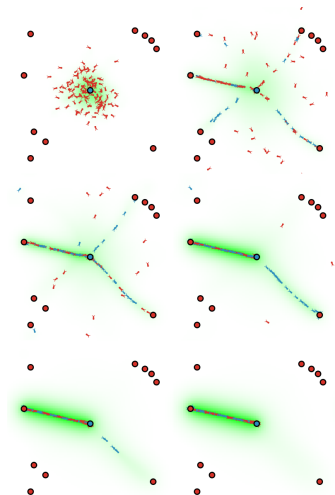


РИС. 6.7 Консенсус в муравьиной колонии

**3** Модифицируйте модель, чтобы начальный уровень феромона в патчах задавался случайным образом, например из диапазона от 0.01 до 0.02. В этом случае поведение модели, особенно на начальном этапе, будет существенно отличаться от случая с нулевым начальным уровнем феромона. В частности, с большой вероятностью будут образовываться ложные цели (рис. 6.8), которые, однако, будут постепенно исчезать благодаря испарению феромона и снижению феромонной активности муравьев в окрестности этих «целей».

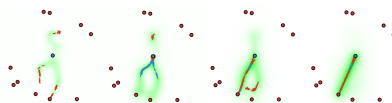


РИС. 6.8 Случайная инициализация феромона

**4** Найденные муравьями маршруты от муравейника к источникам пищи имеют тенденцию к постепенному спрямлению (рис. 6.9). Разработайте эксперимент, который наглядно продемонстрировал бы этот процесс.

Для этого при инициализации модели проложите криволинейную феромонную дорожку от муравейника к источнику пищи, чтобы муравьи сразу выбрали этот маршрут и постепенно его оптимизировали.

**5** Модифицируйте модель с учетом циклического замыкания границ. Для этого перед перемещением муравья командой `fd ant-vel` нужно проверить с помощью команды `patch-ahead ant-vel`, возможно ли это движение: если перед муравьем граница (которую он не может пересечь), то данная команда вернет в качестве результата значение `nobody`. Если движение вперед невозможно, то нужно сменить направление, например случайным образом.

**6** В природе многие виды муравьев используют только один вид феромона, которым они помечают свой путь при возвращении в муравейник от источника пищи. Для определения направления, в котором находится муравейник, муравьи используют несколько методов. Если источник пищи находится недалеко от гнезда, то используются визуальные ориентиры. Если же источник пищи лежит на большом расстоянии от муравейника, то муравьями применяется техника, называемая *path integration*, которая заключается в том, что муравей в каждый момент времени отслеживает направление на муравейник<sup>3</sup>. Измените описанную в главе модель, чтобы муравьи, обнаружив цель, начинали бы двигаться прямо к муравейнику, т.е. без учета феромона `ph-n` (рис. 6.10).

**7** Разработайте модель, в которой у муравьев нет целей (ни муравейника, ни источников пищи). При движении муравьи используют только один вид феромона, которым они помечают свой путь и на основании концентрации которого они строят каждое свое перемещение. В такой модели муравьи достаточно быстро самоорганизуются в форме одной или нескольких мельниц смерти (рис. 6.11).

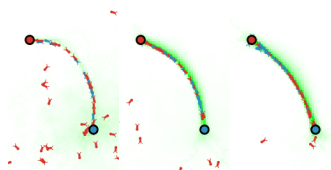


РИС. 6.9 Спрямление пути

<sup>3</sup> M. Muller, R. Wehner, *Path integration in desert ants, Cataglyphis fortis*, Proc. Nati. Acad. Sci. USA, 1988, Vol. 85, p. 5287–5290.

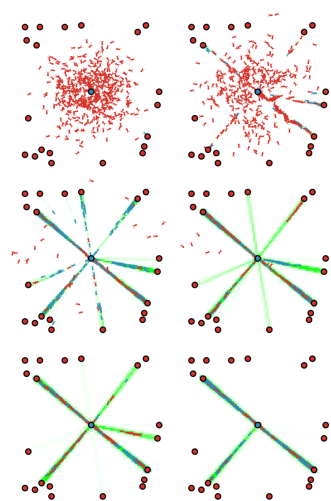


РИС. 6.10 Модель с прямым возвращением в муравейник

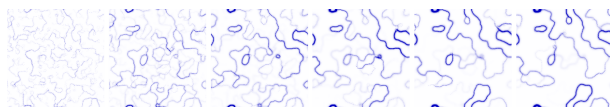
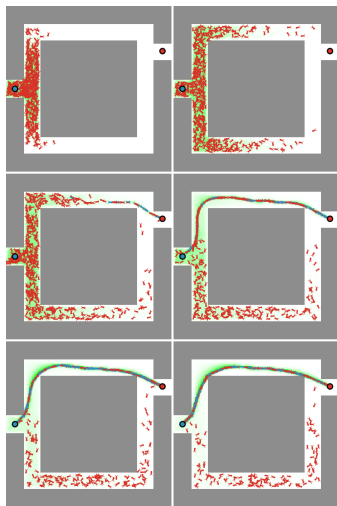


РИС. 6.11 Модель с одним феромоном

**8** Включите в модель поддержку препятствий в форме закрытых для прохода патчей. Рассмотрите поведение колонии муравьев для разных типов карт местности, например которые мы использовали в модели бактериального поиска.



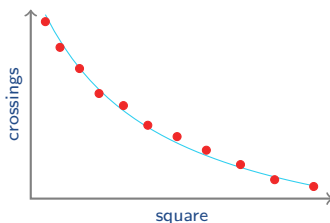
**РИС. 6.12** Эксперимент с двумя мостами, патчи, закрытые для прохода, окрашены в серый цвет

<sup>4</sup> В данной модели феромон не испаряется и не подвергается диффузии.

<sup>5</sup> Т.е. фактически, сколько раз он пересекает свой первый маршрут.

**9** Сконструируйте карту, в которой между муравейником и единственным источником пищи имеется ровно два пути. Проведите численный эксперимент по проверке гипотезы, что муравьи с большей вероятностью выбирают более короткий путь к источнику пищи. Пример работы такой модели приведен на рис. 6.12.

**10** Разработайте модель для упражнения 10 из первой главы, с помощью которой можно было бы численно проверить описанную в этом упражнении способность муравьев оценивать площадь закрытого помещения. В этой модели один муравей делает два прохода одинаковой продолжительности внутри заданной области. При первом проходе он помечает свой путь феромоном<sup>4</sup>. При втором проходе муравей считает, сколько раз он попадает на помеченные патчи<sup>5</sup>. Постройте график зависимости числа пересечений от площади области и убедитесь, что имеется обратная пропорциональная зависимость между этими величинами (рис. 6.13).



**РИС. 6.13** Зависимость числа пересечений двух траекторий муравья от площади области