

Искандер Ситди́ков @Thoughtteer

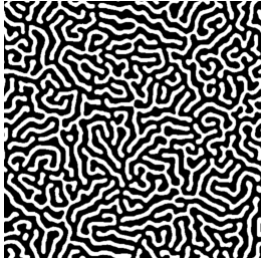
Пользователь

28 мая 2013 в 16:18

Необыкновенный способ генерации лабиринтов

из песочницы

Математика*·Алгоритмы*·Python*








В этой статье я расскажу об одном необычном подходе к генерации лабиринтов. Он основан на модели Амарй нейронной активности коры головного мозга, являющейся непрерывным аналогом нейронных сетей. При определенных условиях она позволяет создавать красивые лабиринты очень сложной формы, подобные тому, что приведен на картинке.

Вас ждет много анализа и немного частных производных. Код прилагается.
Прошу под кат!

Реклама

Х А Б Ы

-  **Информационная безопасность**
2,4k авторов, 6,4k публикаций
-  **Open source**
1k авторов, 2,3k публикаций
-  **Высокая производительность**
546 авторов, 1,2k публикаций
-  **Программирование**
2,9k авторов, 6,5k публикаций
-  **Разработка систем передачи данных**
62 автора, 117 публикаций
-  **Разработка под Linux**
121 автор, 191 публикация
-  **Алгоритмы**
1,3k авторов, 2,3k публикаций
-  **Системное программирование**
218 авторов, 453 публикации
-  **Тестирование веб-сервисов**
201 автор, 296 публикаций
-  **Анализ и проектирование систем**
510 авторов, 974 публикации

Все хабы

Ч И Т А Ю Т С Е Й Ч А С

24-ядерный CPU, а я не могу сдвинуть курсор

31,3k 181

Linux все еще не торт

18,7k 31

Синхронизация Vivaldi: ответы на вопросы

996 5

GitLab CI: ветки больше не нужны

6,2k 8

Эксперимент: возникает ли финансовое неравенство при случайной раздаче денег

6,2k 14

Анализ трафика GSM сетей в Wireshark

4,9k 4

Введение

Многие из читателей уже сталкивались с задачей генерации лабиринта в той или иной форме и знают, что для ее решения зачастую используют алгоритмы Прима и Крускала нахождения минимального остовного дерева в графе, вершины которого являются ячейками лабиринта, а ребра представляют проходы между соседними ячейками. Мы же сделаем смелый шаг прочь от теории графов в сторону... вычислительной нейробиологии.

В течение XX века ученые строили математические модели одиночных нейронов (клеток нервной системы) и их взаимодействия между собой. В 1975 году С. Амарй представил свету свою непрерывную модель коры головного мозга. В ней нервная система рассматривалась как сплошная среда, в каждой точке которой находится «нейрон», характеризующийся значением потенциала своей мембраны, которая меняет свой потенциал, обмениваясь зарядами с соседними нейронами и внешними раздражителями. Модель Амарй знаменита тем, что объясняет многие феномены человеческого зрения и, в частности, зрительные галлюцинации, вызываемые психоактивными веществами.

Модель Амарй, в ее простейшем виде, представляет собой задачу Коши для одного интегро-дифференциального уравнения:

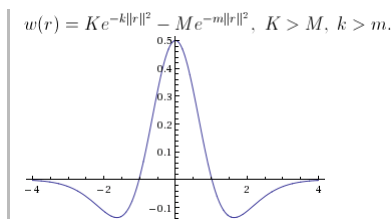
$$\begin{cases} \frac{\partial u}{\partial t}(x, t) = -u(x, t) + h + \int_{\Omega} w(x - y) H(u(y, t)) dy + s(x, t), & x \in \Omega, \quad t > 0, \\ u(x, 0) = \phi(x), & x \in \Omega. \end{cases}$$

Здесь не обойтись без пояснений:

- $u(x, t)$ — вещественное значение потенциала мембраны нейрона в точке x в момент времени t .
- h — потенциал покоя (некоторая вещественная константа).
- H — ступенчатая функция Хэвисайда:

$$H(u) = \begin{cases} 1, & u > 0, \\ 0, & u \leq 0. \end{cases}$$

- w — весовая функция.
- s — внешний раздражитель.
- ϕ — распределение потенциала в начальный момент времени.
- x — произвольная точка области ζ , на которой определен потенциал. Поскольку мы планируем генерировать двумерное изображение лабиринта, в качестве ζ будем рассматривать всю вещественную плоскость.
- Частная производная u по времени в левой части обозначает мгновенное изменение потенциала u . Правая часть задает правило этого изменения.
- Первые два слагаемых правой части означают, что при отсутствии раздражителей значение потенциала стремится к значению потенциала покоя.
- Следующее слагаемое учитывает воздействие соседних нейронов. Функция Хэвисайда играет роль активационной функции нейрона: нейрон начинает влиять на соседей лишь при условии, что его потенциал больше нуля. Будем далее называть такие нейроны активными, а множество точек с положительным потенциалом — областью активности. Ясно, что покоящиеся нейроны не должны быть активными, то есть потенциал покоя не должен быть положительным. Активных соседей можно условно разделить на две группы: возбуждающие и тормозящие. Возбуждающие нейроны увеличивают потенциал соседей, а тормозящие — уменьшают. При этом возбуждающие создают мощный всплеск активности в малой окрестности, а тормозящие постепенно гасят активность в окрестности большого радиуса. Именно этот факт отражен в выборе весовой функции w в форме «мексиканской шляпы»:



- Последнее слагаемое правой части уравнения учитывает действие внешнего раздражителя. Например, для зрительной коры головного мозга естественным раздражителем является сигнал, полученный с сетчатки глаза. Будем считать, что раздражитель задан неотрицательной стационарной (независящей от времени) функцией.

Зададимся вопросом: можно ли подобрать параметры модели ϕ, h, K, k, M, m, s так, чтобы ее стационарное решение (при $t \rightarrow \infty$) было изображением некоторого лабиринта?

Свойства решений модели Амари

Для анализа решений модели Амари нам будет достаточно ограничиться рассмотрением одномерного случая. Для простоты будем полагать, что s постоянна на всей прямой.

В первую очередь нас интересуют так называемые бамп-решения. Они замечательны тем, что положительны лишь на некотором конечном интервале $(-A(t), A(t))$ с подвижными границами.

Уравнение Амари для них записывается следующим образом:

$$\frac{\partial u}{\partial t}(x, t) = -u(x, t) + h + \int_{-A(t)}^{A(t)} w(x-y) dy + s, \quad -\infty < x < \infty, \quad t > 0.$$

Чтобы понять, как ведет себя его решение, введем функцию

$$W(a) = \int_{-a}^a w(x) dx.$$

Теперь то же уравнение можно переписать так:

$$\frac{\partial u}{\partial t}(x, t) = -u(x, t) + h + \frac{1}{2} [W(x+A(t)) - W(x-A(t))] + s.$$

Нам известно, что бамп-решение обращается в ноль на границах интервала активности (потому они и называются границами). Запишем это условие на правой границе:

$$u(A(t), t) = 0, \quad t > 0.$$

А теперь продифференцируем последнее тождество по переменной t :

$$\frac{du}{dt}(A(t), t) = \frac{\partial u}{\partial x}(A(t), t)A'(t) + \frac{\partial u}{\partial t}(A(t), t) = 0.$$

Отсюда:

$$\frac{\partial u}{\partial t}(A(t), t) = -\frac{\partial u}{\partial x}(A(t), t)A'(t).$$

Подставляя последнее выражение в уравнение для бамп-решения при $x = A(t)$, получим:

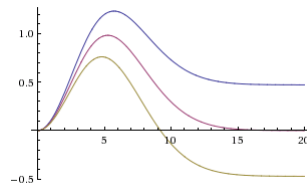
$$-\frac{\partial u}{\partial x}(A(t), t)A'(t) = h + \frac{1}{2}W(2A(t)) + s, \quad t > 0.$$

Теперь заметим, что частная производная по x в левой части всегда отрицательна, так как слева от правой границы решение больше нуля, а справа от нее — меньше. Поэтому

$$\operatorname{sgn} A'(t) = \operatorname{sgn} \left\{ h + \frac{1}{2}W(2A(t)) + s \right\}.$$

Таким образом, направление сдвига границы зависит лишь от значения выражения в правой части. Если оно больше нуля, то область активности расширяется, если меньше — сужается. При равенстве нулю достигается равновесие.

Взглянем на возможные графики функции $W(a)$.



Очевидно, возможны два случая:

1. Предельное значение $h + \frac{1}{2}W(\infty) + s$ неотрицательно. Тогда область активности бамп-решения будет неограниченно расширяться.
2. Предельное значение $h + \frac{1}{2}W(\infty) + s$ отрицательно. Тогда область активности будет ограничена.

Более того, в этом случае можно показать, что **связные компоненты области активности решения уравнения Амари никогда не сливаются**.

К сожалению, в двумерном случае получить явное выражение для функции $W(a)$ затруднительно, поэтому мы просто оценим ее:

$$W(a) \approx \int_{||x|| < a} w(x) dx = \pi \left(\frac{K}{k} (1 - e^{-ka^2}) - \frac{M}{m} (1 - e^{-ma^2}) \right).$$

Отсюда:

$$W(\infty) \approx \pi \left(\frac{K}{k} - \frac{M}{m} \right).$$

Генерация лабиринта

Собрав багаж необходимых знаний, мы можем приступить к, собственно, алгоритму генерации лабиринта.

Прежде всего, определимся с самим понятием «лабиринт». Под лабиринтом будем подразумевать бинарную функцию $L: \Omega \rightarrow \{0, 1\}$ такую, что область $\{x \in \Omega: L(x) = 0\}$ связна. Значение 0 соответствует свободной ячейке, а значение 1 — непроходимой стене. Условие связности говорит о том, что из любой свободной ячейки можно добраться до любой другой, не разрушая стены. Функцию L будем искать в виде:

$$L(x) = \begin{cases} 0, & u(x, \infty) \leq 0, \\ 1, & u(x, \infty) > 0, \end{cases}$$

где u — решение модели Амари. Осталось лишь определиться с параметрами модели.

Начнем с того, что зафиксируем произвольное отрицательное значение h . Естественнo положить $\phi \equiv h$. Теперь зададим функцию s . Пусть ее значение в каждой точке определяется случайной величиной,

равномерно распределенной на отрезке $[0, -h]$. В таком случае раздражитель не будет создавать активность. Зафиксируем произвольное положительное K . Этот параметр влияет лишь на абсолютную величину потенциала, потому не представляет интереса. Зафиксируем произвольные положительные $k, m: k > m$. Они определяют характерную толщину стен лабиринта. Параметр M попробуем определить экспериментально, а затем сравнить с теоретической оценкой, полученной в предыдущем разделе.

Стационарное решение будем искать методом последовательных приближений:

$$\left\{ \begin{array}{l} u = \lim_{k \rightarrow \infty} u_k, \\ u_k(x) = h + \int_{\Omega} w(x-y) H(u_{k-1}(y)) dy + s(x), \quad x \in \Omega, \\ u_0 = \phi. \end{array} \right.$$

А вот и долгожданная интерактивная демонстрация на Python:

```

import math
import numpy
import pygame
from scipy.misc import imsave
from scipy.ndimage.filters import gaussian_filter

class AmariModel(object):

    def __init__(self, size):
        self.h = -0.1
        self.k = 0.05
        self.K = 0.125
        self.m = 0.025
        self.M = 0.065

        self.stimulus = -self.h * numpy.random.random(size)
        self.activity = numpy.zeros(size) + self.h
        self.excitement = numpy.zeros(size)
        self.inhibition = numpy.zeros(size)

    def stimulate(self):
        self.activity[:, :] = self.activity > 0

        sigma = 1 / math.sqrt(2 * self.k)
        gaussian_filter(self.activity, sigma, 0, self.excitement, "wrap")
        self.excitement *= self.K * math.pi / self.k

        sigma = 1 / math.sqrt(2 * self.m)
        gaussian_filter(self.activity, sigma, 0, self.inhibition, "wrap")
        self.inhibition *= self.M * math.pi / self.m

        self.activity[:, :] = self.h
        self.activity[:, :] += self.excitement
        self.activity[:, :] -= self.inhibition
        self.activity[:, :] += self.stimulus

class AmariMazeGenerator(object):

    def __init__(self, size):
        self.model = AmariModel(size)

        pygame.init()
        self.display = pygame.display.set_mode(size, 0)
        pygame.display.set_caption("Amari Maze Generator")

    def run(self):
        pixels = pygame.surfarray.pixels3d(self.display)

        index = 0
        running = True
        while running:
            self.model.stimulate()

            pixels[:, :, :] = (255 * (self.model.activity > 0))[:, :, None]
            pygame.display.flip()

            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    running = False
                elif event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_ESCAPE:
                        running = False
                    elif event.key == pygame.K_s:
                        imsave("{}0:04d}.png".format(index), pixels[:, :, 0])
                        index = index + 1
                    elif event.type == pygame.MOUSEBUTTONDOWN:
                        position = pygame.mouse.get_pos()
                        self.model.activity[position] = 1

            pygame.quit()

    def main():
        generator = AmariMazeGenerator((512, 512))
        generator.run()

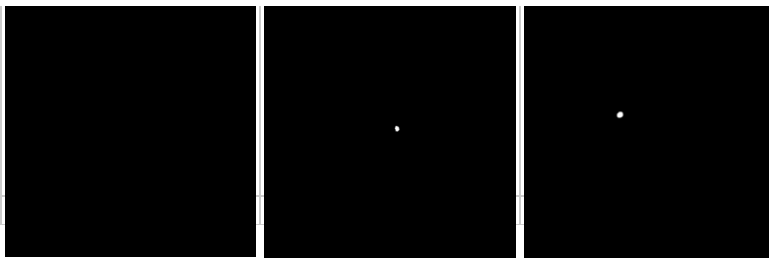
if __name__ == "__main__":
    main()

```

Я полагаю, что комментарии излишни. Хотелось бы отметить лишь то, что свертка с весовой функцией вычисляется через фильтр Гаусса, причем изображения продолжаютсЯ периодически на всю плоскость (параметр «wrap»). Демонстрация интерактивна в том смысле, что позволяет принудительно установить положительный потенциал в любой точке по клику.

Поведение решения, как и ожидалось, зависит от выбора параметра M :

--	--	--



Теперь получим теоретическую оценку оптимального значения параметра M . Оно удовлетворяет условию:

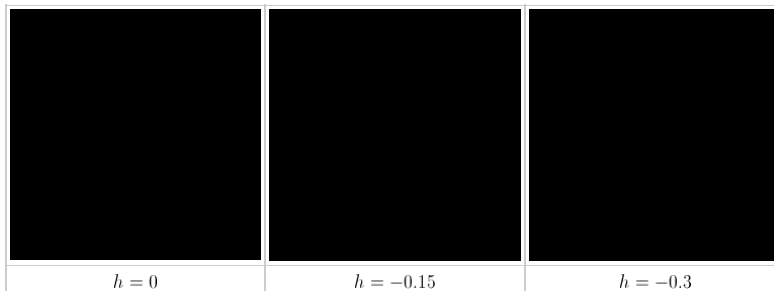
$$h + \frac{1}{2}W(\infty) + \max_{x \in \Omega} s(x) = 0 \Leftrightarrow W(\infty) = 0.$$

Поэтому его можно оценить следующим образом:

$$W(\infty) \approx \pi \left(\frac{K}{k} - \frac{M}{m} \right) \approx 0 \Rightarrow M \approx m \frac{K}{k} = 0.0625.$$

Неплохо, однако реальное значение M чуть выше теоретической оценки. В этом легко убедиться, положив $h = 0$.

Наконец, можно менять степень «разреженности» лабиринта, изменяя значение параметра h :



Заключение

Вот мы и закончили рассмотрение, пожалуй, самого необыкновенного способа генерации лабиринтов. Надеюсь, что статья показалась Вам интересной. В заключение приведу список литературы для желающих расширить свой кругозор:

- [1] Konstantin Doubrovinski, *Dynamics, Stability and Bifurcation Phenomena in the Nonlocal Model of Cortical Activity*, 2005.
- [2] Dequan Jin, Dong Liang, Jigen Peng, *Existence and Properties of Stationary Solution of Dynamical Neural Field*, 2011.
- [3] Stephen Coombes, Helmut Schmidt, Ingo Bojak, *Interface Dynamics in Planar Neural Field Models*, 2012.

генерация лабиринтов, нейронные сети, модель Амары

Сильный отворот, кратчайшие сроки ×
amun-mag.ru

Прогнозы от экспертов и СМИ ×
4teller.com

Информация про недвижимость, доллар, пенсии и другая! Читайте на 4teller.com

Эксперты
Последние прогнозы
Как стать экспертом
Частые вопросы


Услуги заполнения документов ×
заполнение.дм-консалтинг.рф

Услуги **заполнения** документов на заказ. Коллектив специалистов.

Адрес и телефон

Яндекс.Директ
Реклама

↑ +263 ↓ 👁 74,9k ★ 882 🐦 👤 📘 ❤

 **Искандер Ситди́ков** @Thoughteer карма 83,0 рейтинг 0,0

ПОХОЖИЕ ПУБЛИКАЦИИ

12 февраля в 18:38
Нейронные сети для начинающих. Часть 2
↑ +38 👁 80,3k ★ 722 💬 30

7 декабря 2016 в 13:21
Нейронные сети на JS. Создавая сеть с нуля
↑ +49 👁 51,6k ★ 481 💬 42

10 ноября 2016 в 17:35
Deep Learning: Transfer learning и тонкая настройка глубоких сверточных нейронных сетей
↑ +21 👁 10,2k ★ 101 💬 0

САМОЕ ЧИТАЕМОЕ Разработка

Сутки Неделя Месяц

24-ядерный CPU, а я не могу сдвинуть курсор
↑ +116 👁 31,3k ★ 103 💬 181

Linux все еще не торт			
↑ +75	👁 18,7k	★ 59	💬 31
Первая российская материнская плата массового сегмента			
↑ +55	👁 27,2k	★ 47	💬 193
Как правильно использовать переменные в препроцессорах LESS и SASS			
↑ +19	👁 12,1k	★ 99	💬 5
Уязвимость ВКонтакте: отправляем сообщение с кодом восстановления страницы на чужой номер			
↑ +216	👁 47,6k	★ 187	💬 123

<div>👤</div> <div>safright 28 мая 2013 в 16:22 #</div> <div>Просто потрясающе!</div>		+16 ↑ ↓
<div>👤</div> <div>stg34 28 мая 2013 в 16:24 #</div> <div>Нужна ли математика программисту?</div>		+45 ↑ ↓
<div>👤</div> <div>Niktz 28 мая 2013 в 16:32 # h i</div> <div>Suicide mode ON</div>		-23 ↑ ↓
<div>👤</div> <div>vvzvlad 29 мая 2013 в 15:31 # h i</div> <div>И действительно.</div>		+2 ↑ ↓
<div>👤</div> <div>ideator 28 мая 2013 в 16:34 # h i</div> <div>Нужен ли программист математике?</div>		+29 ↑ ↓
<div>👤</div> <div>Prototik 28 мая 2013 в 17:39 # h i</div> <div>Нужен ли программист?</div>		+17 ↑ ↓
<div>НЛО прилетело и опубликовало эту надпись здесь</div>		
<div>👤</div> <div>tirinox 28 мая 2013 в 22:14 # h i</div> <div>Нужен!</div>		-3 ↑ ↓
<div>👤</div> <div>Siorinex 28 мая 2013 в 22:31 # h i</div> <div>«Алло, Китайцы? У вас есть миллиард? Здорово! А не могли-бы вы мне его прислать?» — Лукашенко из Мультя Личности.</div>		-2 ↑ ↓
<div>👤</div> <div>AlexeyK 28 мая 2013 в 22:52 # h i</div> <div>Не трогайте Брюса.</div>		-2 ↑ ↓
<div>//</div> <div>anarleen 29 мая 2013 в 12:12 # h i</div> <div>Как минимум — Софус.</div>		+1 ↑ ↓
<div>👤</div> <div>daspisch 28 мая 2013 в 16:50 # h i</div> <div>Ответ: Да, в частных случаях. В других частных случаях программисту нужна биохимия, экономика и куча других знаний необходимых для конкретного проекта. Реально достали уже с этим «нужналипрограммистуматематика».</div>		+56 ↑ ↓
<div>👤</div> <div>КванТТТ 28 мая 2013 в 20:11 # h i</div> <div>Нужен ли этот вопрос здесь?</div>		+5 ↑ ↓
<div>👤</div> <div>Aux 28 мая 2013 в 20:53 # h i</div> <div>Нет, есть гугл.</div>		-7 ↑ ↓
<div>👤</div> <div>anigillator 29 мая 2013 в 03:40 # h i</div> <div>Программист — он как профессор Мориарти — по определению математик!</div>		+1 ↑ ↓
<div>👤</div> <div>silvansky 28 мая 2013 в 16:31 #</div> <div>Это офигенно!</div>		+7 ↑ ↓
<div>Респект как от лица программиста, так и от лица математика!</div>		
<div>👤</div> <div>spacediver 29 мая 2013 в 00:45 # h i</div>		+11 ↑ ↓

Ч Т О О Б С У Ж Д А Ю Т	
Сейчас	Вчера Неделя
Первая российская материнская плата массового сегмента	
👁 27,2k	💬 193
24-ядерный CPU, а я не могу сдвинуть курсор	
👁 31,3k	💬 181
Линейное программирование в python силами библиотеки scipy	
👁 1,7k	💬 5
ПЛК от производителей Овен, Segnetics и Schneider Electric для HVAC	
👁 3,2k	💬 50
Анализ трафика GSM сетей в Wireshark	
👁 4,9k	💬 4



 printf 28 мая 2013 в 16:59 #


+3 ↑ ↓

Очень классно, спасибо. Иллюстрации отличные.

 moov 28 мая 2013 в 17:06 #

+5 ↑ ↓

Это просто великолепно! Автор — молодец, Хабр — торт!

 kaichou 28 мая 2013 в 17:21 #

+3 ↑ ↓

Круто. Ещё решить бы, как добиться решаемости сгенерированного лабиринта by design. Не «пробовать» же «дырки» в «стенах», попутно натравливая на него всякие A*.

 Thoughteer 28 мая 2013 в 17:56 # h i

+3 ↑ ↓

Этому вопросу посвящен целый раздел.

Попробую объяснить на пальцах: если запускать демо с теми параметрами, что указаны в посте ($M = 0.065$), то белые «извилины» (стены) никогда не будут сливаться. Поскольку изначально черная область была связана (из любой черной точки можно было дойти до любой другой черной точки, передвигаясь лишь по черным точкам), то и в любой другой момент времени она останется связанной. Иначе говоря, решаемости лабиринта «by design» можно добиться правильным выбором параметров модели (точнее, одного параметра M).

 kaichou 28 мая 2013 в 18:41 # h i

0 ↑ ↓

Спасибо за подробный ответ!

 andreishe 28 мая 2013 в 19:57 # h i

+2 ↑ ↓

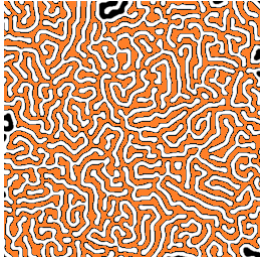
Поскольку изначально черная область была связана (из любой черной точки можно было дойти до любой другой черной точки, передвигаясь лишь по черным точкам), то и в любой другой момент времени она останется связанной.

Только эта «связность» будет обеспечиваться наличием общей «внешней» оболочки, что несколько странно для лабиринта: они обычно обнесены стеной по периметру.

Вот немного пейнта по упомянутой картинке из статьи:



Хотя вот здесь все гораздо лучше:



 Thoughteer 28 мая 2013 в 22:11 # h i

+1 ↑ ↓

Все абсолютно верно. В представленном варианте связным будет лишь бесконечное периодическое продолжение лабиринта на всю плоскость. Это связано с использованием режима «warp» в фильтре Гаусса. Для того, чтобы получить **конечный** связный лабиринт, достаточно поменять «warp» на «reflect».

 Thoughteer 28 мая 2013 в 22:16 # h i

+1 ↑ ↓

Вот пример для наглядности:



NetBUG 29 мая 2013 в 10:31 # h i

0 ↑ ↓

Вот интересны границы интервала значений M, когда лабиринт будет терять связность либо вырождаться в белое либо чёрное пятно.

Надо поэкспериментировать и потом попробовать доказать.

spiritedflow 28 мая 2013 в 19:33 # h i

+1 ↑ ↓

Тут скорее нужно не ломать стенки, а, наоборот, закрывать лишние дырки. Например, на самой первой картинке вариантов прохода от одной стены к другой слишком много. Вот тогда будет полная жесь :)

meteozone 28 мая 2013 в 17:22 #

0 ↑ ↓

вот это хардкор!

SamSol 28 мая 2013 в 17:23 #

+13 ↑ ↓

О, извилины! Дефицит! Мне такие нужны!

sysprg 28 мая 2013 в 17:24 #

0 ↑ ↓

Спасибо, красиво! Но что касается практичности, то главное свойство лабиринта – запутанность. В нем должен быть неочевидный правильный путь (или небольшое количество правильных путей, не считая петель и циклов), плюс большое количество тупиков или петель. А ваши лабиринты проходятся почти по прямой. Можно ли попростому доработать алгоритм так, чтобы удлинить и запутать правильный путь?

Thoughteer 28 мая 2013 в 17:41 # h i

+2 ↑ ↓

Согласен, но, боюсь, простейшей модификацией не обойтись. Скажем, это не те лабиринты, в которых есть «правильный путь», а те лабиринты, которые можно использовать в гейм-дизайне для генерации интересных уровней.

gleb_kudr 28 мая 2013 в 17:30 #

+1 ↑ ↓

Вы довольно точно смоделировали рост некоторых колоний бактерий, или, например, коралла-мозговика :)

Daymz 28 мая 2013 в 17:44 #

+15 ↑ ↓



```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

[Видео](#)

[Книга](#)

[Обзор](#)

silvansky 28 мая 2013 в 18:04 # h i

+6 ↑ ↓

Никто не может гарантировать связность этого лабиринта.

ariel32 28 мая 2013 в 18:57 #

+5 ↑ ↓

Заглянул под кат с мыслью: «Что может быть *необыкновенного* в генерации лабиринта?», а тут *такое*... Снимаю шляпу, вы украли у меня часов 5 ночного сна!
"ушел генерировать лабиринты"

Aingis 29 мая 2013 в 14:39 # h i

-3 ↑ ↓

Хм-м, время: 18:57, место: Москва, если верить профилю. Вы спите днём?

Shultc 30 мая 2013 в 04:08 # h i

0 ↑ ↓

... человек просто спит не дома. Только тсссс! Жене его не говорите!

kliss 28 мая 2013 в 19:41 #

+4 ↑ ↓

Одна из тех очень немногих статей, ради которых я до сих пор и читаю хабр. По пальцам можно пересчитать их.

erley 28 мая 2013 в 22:00 #

0 ↑ ↓

Благодарю за потрясающую иллюстрацию силы математики!

guyfawkes 28 мая 2013 в 22:55 #

0 ↑ ↓

Спасибо за статью! Может быть, один лишь я не все понял, но все же: не могли бы вы пояснить, откуда берутся подобные формулы для сигмы: $\sigma = 1 / \text{math.sqrt}(2 * \text{self.k})$ и $\sigma = 1 / \text{math.sqrt}(2 * \text{self.m})$ и какую роль она выполняет?

Thoughteer 29 мая 2013 в 00:13 # h i

+1 ↑ ↓

В этом месте кода вычисляется *свертка* activity с весовой функцией

$$w(r) = K e^{-kr^2} - M e^{-kr^2}.$$

Реализовывать такую операцию руками мне не хотелось, поэтому я представил ее в виде разности результатов двух сверток: с $K e^{-kr^2}$ и с $M e^{-mr^2}$. В свою очередь, свертка с функцией $K e^{-kr^2}$ по сути есть свертка с функцией Гаусса (с точностью до постоянного множителя):

$$K e^{-kr^2} = C \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \quad \sigma = \frac{1}{\sqrt{2k}}, \quad C = K \frac{\pi}{k}.$$

Отсюда и берутся значения всех таинственных констант.


reinvigorate 29 мая 2013 в 11:54 #

0 ↑ ↓









А мне кажется, или подобный результат возможен вообще без сложной математики?

Вот просто задавать несколько раз N точек, их движение, следить чтобы не сближались, вероятность разветвления, постепенное изменение угла движения.









Кажется очень просто.

 **Suvitruf** 29 мая 2013 в 12:28      +1  









Так не интересно и не элегантно)

 **reinvigorate** 29 мая 2013 в 18:35      0  









Зато так по идее можно гарантировать решаемость алгоритма и задавать его сложность.

 **Shultc** 30 мая 2013 в 04:10      +1  

Напишите алгоритм. Покажите нам. Посмотрим, сравним.

 **romangorbunov** 29 мая 2013 в 13:10      +1  









Мне, как гуманитария, это разрывает мозг. Многое непонятно, но понятно, что это потрясающе. Не смотря на то, что по математике у меня была уверенная пятерка, сложные формулы заставили лицо скривить гримасу ужаса от шока непонимания!

 **valyard** 29 мая 2013 в 13:33      +10  


Статья хорошая, но является классическим примером советского образования.

Сначала идет куча математики, которую даже я, чертов Магистр Математики, вечером за пивком решил пролистать. Хорошо, что не закрыл. Ведь дальше идет как раз интересная часть. И так везде. На протяжении шести лет университета было ровно так же. Три месяца скучнейшей абстрактной математики, которая совершенно не понятно зачем нужна. А к моменту, когда начинается реальное применение, 95% студентов уже отвалилось и ничего не понимают.









Должно быть совершенно наоборот: показать в начале классное применение, упуская объяснение сложных моментов, а уже потом постепенно разворачивать сложность и вводить новые абстракции и формулы.

 **S_A** 29 мая 2013 в 14:17      +1  









Мои 4 курса ММФ кончились 10 лет назад (хоть я все же и Магистр, но Других Наук), но я испытал удовольствие от формул, хотя вообще-то далеко не с первого раза врубился. Суть ведь не в программе, так же? А чтобы самим делать подобное. Абстракция нужна чтобы иметь не единственное применение, и этот пост тому пример. И еще и мотивация.

 **valyard** 29 мая 2013 в 14:35      0  









Вы коммент-то прочитали?

 **S_A** 29 мая 2013 в 14:49      0  









Переформулирую. Мысль в том, что практическая реализация — это дело второго приоритета. Проработать идею важнее.

 **evans2094** 30 мая 2013 в 13:40      0  

К сожалению многие так и остались детьми — им интересно как машинка катается, а не как она устроена что бы кататься. Им это просто скучно и хочется смотреть на катающуюся машинку. Собственно поэтому вам и рекомендуют сначала сделать привлекательную презентацию с картинками и видео для альтернативно одаренных, а потом уже пугать формулами и проработкой идеи. Тем более что модель Коши изучают в технических ВУЗах и фактически здесь мы видим отличный пятерочный курсач второкурсника физтеха или ВМК по мат методам.


 **khddavid** 4 июня 2013 в 11:07      0  





Спасибо за прекрасную статью. Вижу вы мучились с вставкой картинок с формулами. Советую стабильный онлайн сервис их генерации [codecogs](#)





 **artemev** 13 февраля 2014 в 12:30      -3  





Программирование великая вещь! Всего пару десятков строк кода и получается такая красотища. Мне этот метод генерации лабиринтов очень бы пригодился в одном проекте, но с python-ом я, к сожалению, не знаком. Если кто-нибудь возьмется портировать этот код на actionscript 3, то я буду не только благодарен, но и оплачу труды в размере 1000 руб. Кого это предложение интересует, пишите в личку.





Только [полноправные пользователи](#) могут оставлять комментарии. [Войдите](#), пожалуйста.





ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ 


Белые списки. Хорошее начало — половина дела GT
 +5  971  1  7


Качество продуктов упало. Почему? Что делать? GT
 +5  940  2  1

Синхронизация Vivaldi: ответы на вопросы
 +9  996  1  5



«Ген Химеры». Глава 19 GT
 +6  356  0  1

Запуск Java классов и JAR-ов не по учебнику
 +10  2,1k  16  11

Купить орхидеи в Уфу 



Яндекс.Директ
Реклама

Аккаунт	Разделы	Информация	Услуги	Приложения
Войти	Публикации	О сайте	Реклама	 
Регистрация	Хабы	Правила	Тарифы	
	Компании	Помощь	Контент	
	Пользователи	Соглашение	Семинары	
	Песочница	Конфиденциальность		



© 2006 – 2017 «ТМ»

Служба поддержки

Мобильная версия

