

Хамитов М.А. Алгоритмы генерации лабиринтов «Sidewinder» и «Олдоса-Бродера»// Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №7 (июль). – АРТ 444-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.021

Хамитов Марат Амирович

студент 4 курса, факультет информатики и робототехники

Научный руководитель: Атнабаев А. Ф., к.т.н., доцент кафедры ГИС
ФГБОУ ВО «Уфимский государственный авиационный технический

университет»

г. Уфа, Российская Федерация

e-mail: office@ugatu.su

**АЛГОРИТМЫ ГЕНЕРАЦИИ ЛАБИРИНТОВ «SIDEWINDER» И
«ОЛДОСА-БРОДЕРА»**

Аннотация: В статье рассмотрены алгоритмы генерации лабиринтов «Sidewinder» и «Олдоса-Бродера», представлена блок-схема алгоритма «Sidewinder», а также проведен разбор этих алгоритмов.

Ключевые слова: алгоритм, генерация, лабиринт, алгоритм Sidewinder, алгоритм Олдоса-Бродера.

Khamitov Marat

4nd year student, faculty of informatics and robotics

Supervisor: A. Atnabaev, CTS, Associate Professor of Department of GIS
FGBOU VO "Ufa state aviation technical university"

Ufa, Russian Federation

ALGORITHMS FOR LABYRINTH GENERATION «SIDEWINDER» AND «ALDOUS-BRODER»

Abstract: In the article algorithms for generation of labyrinths "Sidewinder" and "Aldous-Broder" are considered, a block diagram of the algorithm "Sidewinder" is presented, as well as a review of these algorithms.

Keywords: algorithm, generation, labyrinth, algorithm Sidewinder, algorithm of Aldous-Broder.

Лабиринтом является какая-либо структура в двумерном, либо трехмерном пространстве, которая состоит из запутанных путей к выходу из него.

В древнем Египте лабиринты применяли для того, чтобы оградить фараонов от внешнего мира, для обеспечения их безопасности. Таким образом они могли управлять страной из центра лабиринта без каких-либо опасений.

В настоящее время лабиринты чаще всего применяются при разработке компьютерных игр. В них, обычно, строятся специальные сети пещер, тоннелей, выбраться из которых не просто. Чем сложнее лабиринт – тем дольше игрок потратит времени на поиск пути, а также исследует больше местности.

Как правило лабиринты строятся не людьми, а при помощи процедурной генерации, используя специальные алгоритмы. Таких алгоритмов насчитывается большое количество. Из них выделяются алгоритмы «Sidewinder», основанный на алгоритме двоичного дерева, а также алгоритм «Олдоса-Бродера», который генерирует максимально

запутанный лабиринт, за счёт большой доли применения «случайностей» в алгоритме.

Алгоритм «Sidewinder»

Алгоритм «Sidewinder» основан на алгоритме двоичного дерева, где путь прокладывается только в двух направлениях. Под прокладыванием пути подразумевается разрушение стенки между клетками на поле, состоящая из множества клеток и стенками между ними. В алгоритме «Sidewinder» используется тот же принцип выбора одного из двух решений, но при этом выбор совершается между «добавить в некоторый массив еще одну клетку ряда» и «выбрать из этого массива одну из клеток и соединить этот массив клеток с верхним рядом через выбранную клетку». Так или иначе, алгоритм является простым для понимания и позволяет создавать бесконечные лабиринты, при этом рисунок лабиринта будет в достаточной степени сложным.

На рисунке 1 представлена блок-схема алгоритма «Sidewinder»:

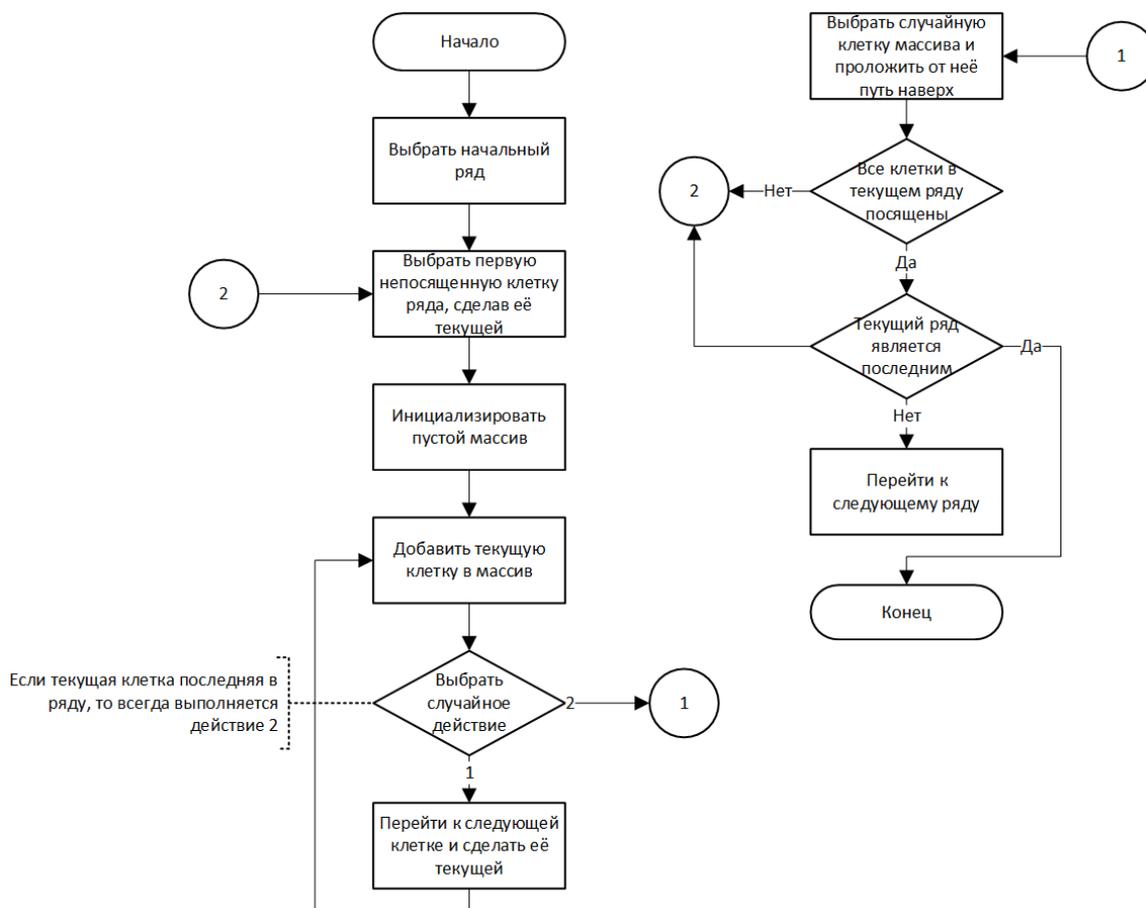


Рисунок 1 – Блок схема алгоритма генерации лабиринта «Sidewinder»

Важным моментом алгоритма является очищение первого ряда от стенок. Таким образом начальным рядом считается ряд, следующий за самым первым.

На рисунке 2 графически представлен разбор алгоритма на примере поля размером 6х6.

Поле №1 отображает процесс очистки первого ряда от стенок. На полях №2-3 представлено дважды выбранное действие «добавить в массив еще одну клетку ряда». На поле №4 производится выбор случайной клетки из массива, после чего все стенки между клетками массива, а также верхняя стенка случайно выбранной клетки уничтожаются. На поле №5

представлена инициализация массива, добавления первой клетки в массив, после чего переход к действию «выбрать из массива одну из клеток и соединить этот массив клеток с верхним рядом через выбранную клетку». В данной ситуации разрушается только верхняя стенка над текущей клеткой. Поля №6-8 отображают тот же процесс, что и поля №2-4.

Результат выполненного алгоритма представлен на поле №9.

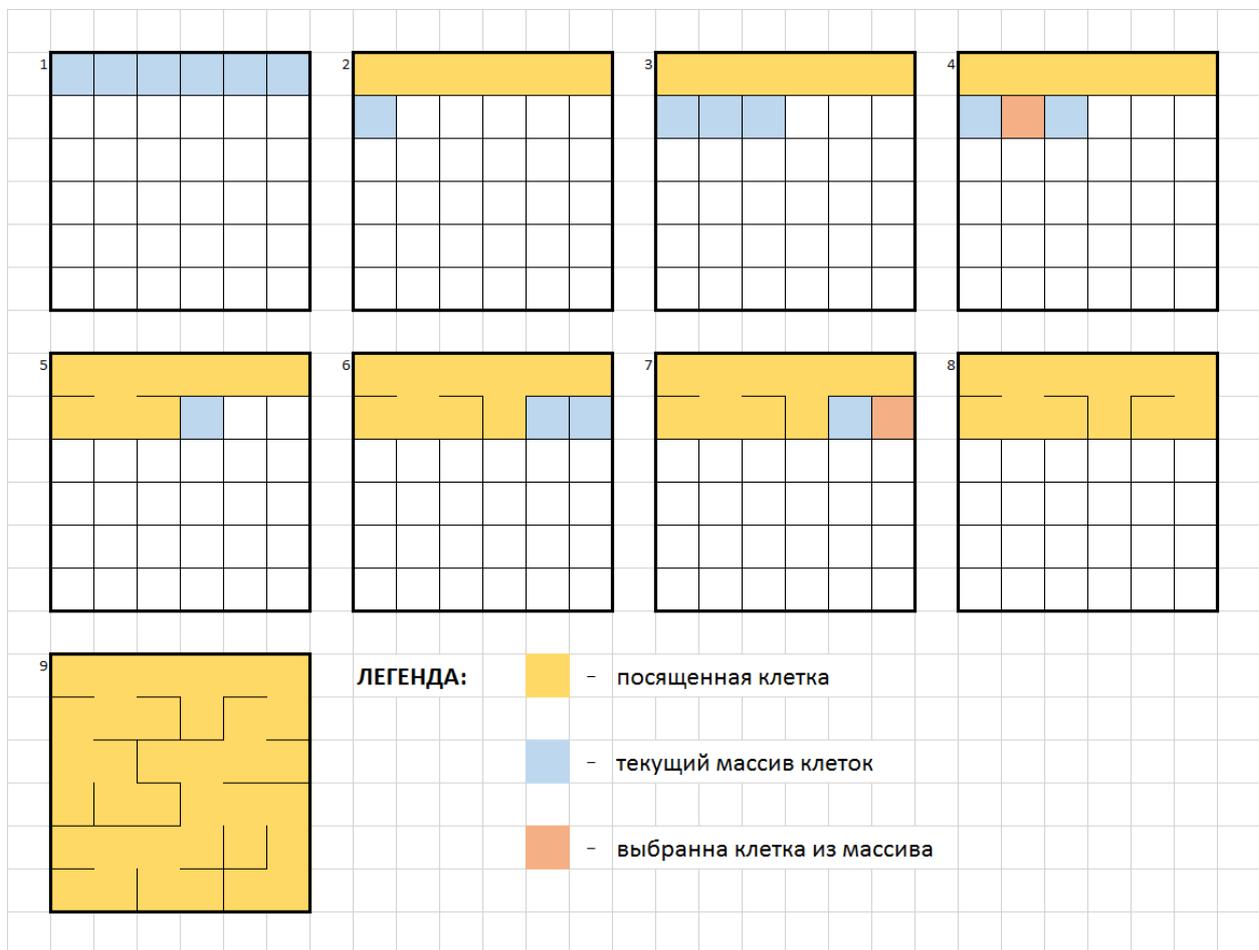


Рисунок 2 – Поэтапный разбор алгоритма генерации лабиринта
«Sidewinder»

Алгоритм «Олдоса-Бродера»

Алгоритм «Олдоса-Бродера» является максимально простым и создает максимально запутанные лабиринты. Объясняется это тем, что в алгоритм состоит трёх действий:

- 1) Выбрать случайную клетку на поле
- 2) Перейти к любой соседней клетке
- 3) Если данная клетка не была ранее посещена, то разрушить между клетками стенку и перейти к действию 2, иначе просто перейти к действию 2.

Таким образом создается максимально непредсказуемый лабиринт. Самым большим минусом данного алгоритма является время построения лабиринта. Даже не очень большой лабиринт может строиться довольно долго.

На рисунке 3 графически представлен разбор алгоритма на примере поля размером 6х6.

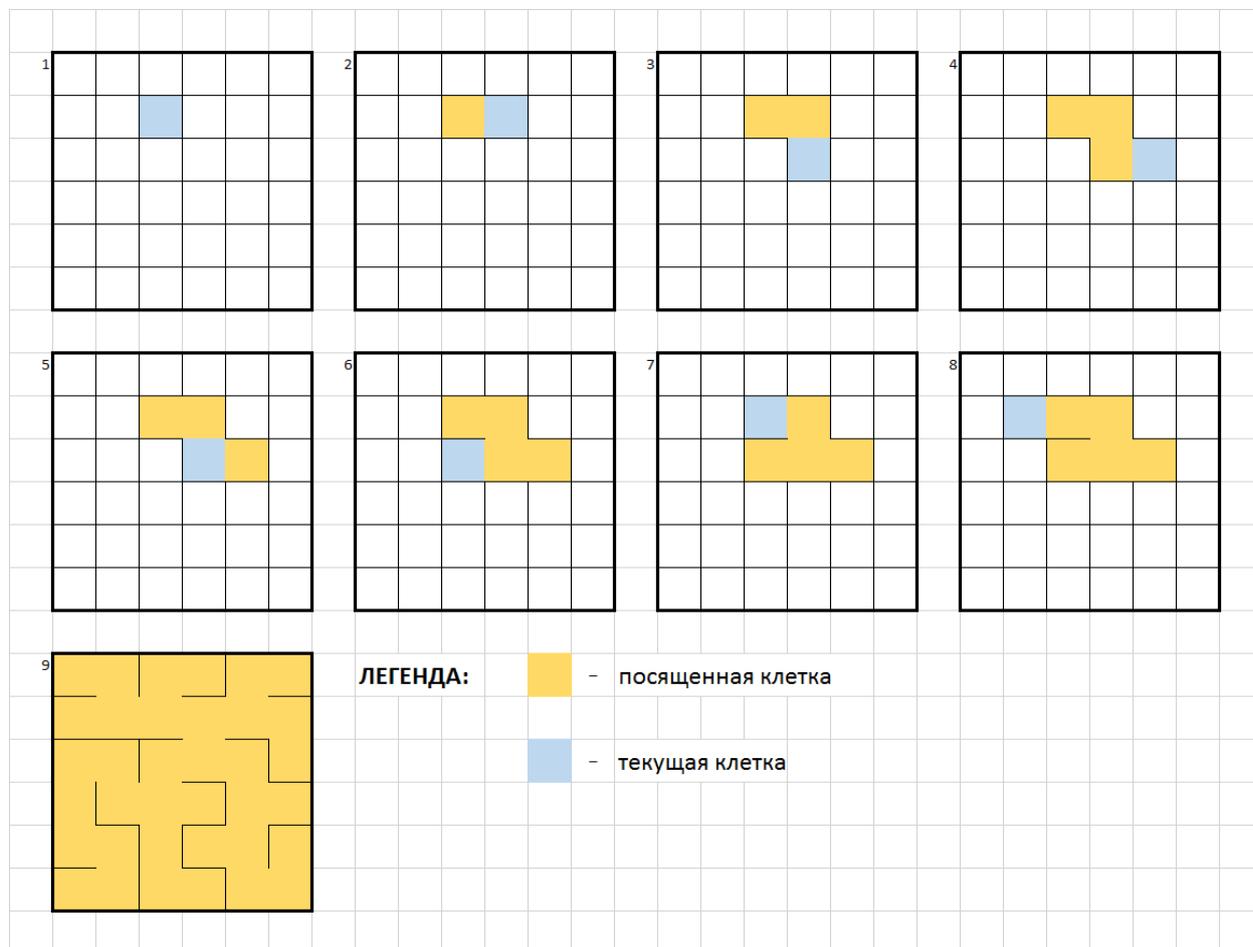


Рисунок 3 – Поэтапный разбор алгоритма генерации лабиринта «Олдоса-Бродера»

На полях №1-4 представлены переходы к клеткам в случайных направлениях, параллельно разрушая стенки. На поле №5 производится переход к уже посещённой клетке. Также подобный переход наблюдается на поле №7, но без разрушения стенки, поскольку переход производится на ранее посещённую клетку. Конечный результат алгоритма представлен на поле №9.

Заключение

Все алгоритмы генерации лабиринтов имеют собственную ценность, и выбор конкретного алгоритма ограничивается лишь потребностью и ресурсами. Алгоритм «Sidewinder» генерирует не самые сложные лабиринты, но выполняет это очень быстро. В случае алгоритма «Олдоса-Бродера» генерируется сложный лабиринт, но при этом в среднем затрачивается много временного ресурса, при чем предсказать затрачиваемое время невозможно, поскольку алгоритм может обойти всё поле, ни разу не перейдя на уже посещённую клетку, а также может больше часа переходить от одной клетки к другой, тем самым посетив всего лишь две клетки.

Список использованной литературы:

1. Классические алгоритмы генерации лабиринтов. Часть 1 // Хабр: веб-сайт в формате коллективного блога. 2018. URL: <https://habr.com/post/320140/>, (дата обращения 24.07.2018).
2. Классические алгоритмы генерации лабиринтов. Часть 2 // Хабр: веб-сайт в формате коллективного блога. 2018. URL: <https://habr.com/post/321210/>, (дата обращения 24.07.2018).

Дата поступления в редакцию: 24.07.2018 г.

Опубликовано: 29.07.2018 г.

© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2018

© Хамитов М.А., 2018