

# Поиск пути в 3D-пространстве с учетом динамических объектов

Винокурова Светлана  
Лаборатория систем мультимедиа,  
Марийский технический университет, Йошкар-Ола, Россия  
svetlana\_curls@mail.ru

## Аннотация

В статье рассматривается расширение метода навигационного графа (НГ) для поиска путей в 3D-пространстве с использованием множества НГ, соответствующих каждому объекту, вместо единого графа. Метод позволяет учитывать динамические объекты без лишних вычислительных затрат в рамках единого алгоритма, а также существенно сокращает как объем ручной работы задания НГ, так и общее время работы алгоритма без ущерба для адекватности найденного пути.

**Ключевые слова:** Поиск путей, Динамические объекты, Навигационный граф.

## 1. ВВЕДЕНИЕ

Поиск пути (ПП) – задача нахождения наилучшего, оптимального маршрута между двумя точками пространства. В зависимости от контекста критерии оптимальности могут быть различные, но во всех случаях на найденном пути не должно встречаться препятствий. Важным достоинством системы ПП является возможность учитывать динамические объекты и корректировать найденный путь при необходимости. Динамическим объектом в системе считается объект, который может изменить позицию и угол поворота, появится или исчезнуть в любой момент времени.

ПП - важная часть виртуальной 3D-среды. ПП используется в компьютерных играх, различных симуляторах и тренажерах, чтобы обеспечивать корректное перемещение персонажей, управляемых компьютером. Кроме того ПП применяется в виртуальных мирах для управляемого пользователем персонажа.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Для ПП существуют 4 основных метода.

Первый из них подразумевает разбиение всего пространства на клетки одинаковой формы, для каждой из которых задано, проходима клетка или нет, и поиск пути алгоритмом A\* [6]. Подразумевается, что объект, для которого идет ПП, занимает 1 клетку (Рис.1). Метод успешно используется в 2D-пространстве, но плохо применим к 3D-миру, т.к. имеет в этом случае серьезные недостатки:

- 3D-мир состоит из объектов произвольной формы, которые плохо ложатся на клеточную структуру пространства.
- Трудно учитывать перемещающиеся препятствия, так как при произвольной позиции они могут

занимать до 4 клеток сразу (если размер препятствия 1 клетка).

- Перемещение по клеткам выглядит неестественно в трехмерном пространстве. Методы сглаживания пути сплайнами могут привести к проблемам с непроходимостью отдельных участков пути.
- Большое время ПП при большом количестве клеток.

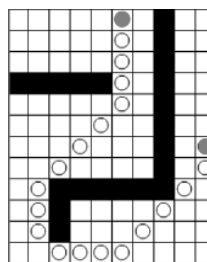


Рис. 1: Пример поиска пути на клеточном пространстве с помощью алгоритма A\*.

Метод НГ предполагает ПП на графе, где вершины графа - трехмерные точки, ребра графа - отрезки, соединяющие эти точки, с ценой равной длине отрезка [2]. Все ребра графа являются проходимыми (персонаж может пройти по ним, не наткнувшись на препятствие). Задача ПП сводится к нахождению ближайших вершин к начальной и конечной точке, а затем к ПП на графе между этими вершинами с использованием критерия минимального веса общего пути (Рис.2). Метод хорошо подходит для 3D-пространства, однако можно принять во внимание 4 основных недостатка метода НГ [5]:

1. Большая трудоемкость задания графа, т.к. в некоторых случаях требуется задать чрезмерное число вершин графа.
2. Перемещаемые при помощи ПП объекты проходят пустое пространство по неестественной траектории, т.к. следуют по заданным ребрам графа. Проявление проблемы можно смягчить, если увеличить число вершин графа, что усугубит недостаток № 1 (Рис.3).
3. В ситуации, когда динамический объект находится на ребре НГ, нет корректного способа построить путь через это ребро.
4. НГ плохо подходит для ПП для перемещающихся объектов разных размеров, т.к. при размещении вершин графа приходится ориентироваться только на 1 конкретный размер персонажа.

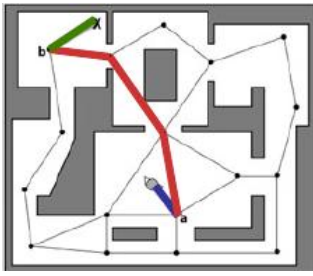


Рис. 2: Пример поиска пути с помощью НГ.

Метод сочетания эвристик [3] в 3D-пространстве предполагает применение специального алгоритма разрешения проблемной ситуации для небольшого числа распространенных случаев необходимости обхода препятствия. При его помощи полноценный ПП организовать невозможно.

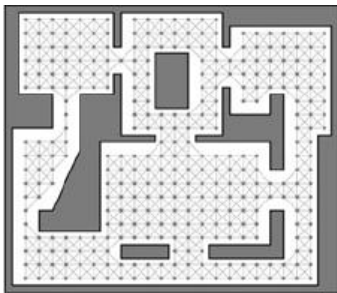


Рис. 3: Увеличение числа вершин графа для улучшения естественности найденного пути.

Метод navigation mesh [4] предполагает задание полигональной 3D-модели проходимого пространства (Рис.4). С помощью этого метода трудно учитывать динамические объекты, так как это требует больших вычислительных затрат.



Рис. 4: Пример navigation mesh.

### 3. ПОИСК ПУТИ НА ОСНОВЕ УСОВЕРШЕНСТВАННОГО МЕТОДА НГ

На основе собственного расширения метода НГ была разработана система ПП для 3D-мира. Для ПП используется множество графов  $A = \{A_i, i = 1..n\}$  (1), где  $A_i = \{V, R\}$  (2) – НГ одного объекта, заданный множеством вершин  $V$  и множеством ребер  $R = \{R_j, j = 1..m\}$  (3), где  $R_j = \{V_{j1} \in V, V_{j2} \in V, T_A, T_D, j1 \neq j2\}$  (4),  $T_A \in \{0,1\}$  – признак

проходимости,  $T_D$  – контекстная информация о проходимости пространства вокруг ребра.

На практике НГ для каждой 3D-модели задается визуально с помощью специального инструмента [8].

ПП выполняется по известному отрезку пути  $S$ , соединяющему начальную и конечную точки  $P_n$  и  $P_k$ . При поиске выполняются следующие шаги:

1. Поиск множества  $A_{изм}$  динамических объектов, у которых изменились угол поворота или позиция. Замена объединенного графа на его составляющие, если найденный объект находился в составе объединенного графа.

$A_{изм} = \{A_i, \text{такие, что } p_i \neq p'_i \text{ или } q_i \neq q'_i\}$  (5), где  $p_i, q_i$  – позиция и угол поворота объекта, соответствующего  $A_i$ , в текущий момент времени, а  $p'_i, q'_i$  – в момент предыдущего ПП.

2. Пересчет координат ребер НГ найденных динамических объектов в соответствии с новой позицией и углом поворота.
3. Поиск множества  $A_{пер}$  динамических объектов, НГ которых пересекаются друг с другом.
4. Объединение пересекающихся НГ динамических объектов и создание временного НГ, который учитывается при ПП вместо пересекающихся.
5. Поиск множества точек пересечения (ТП)  $P$  отрезка пути с НГ объектов. Первоначально выполняется проверка пересечения отрезка пути с bounding box НГ объекта и, только если проверка прошла успешно, выполняется поиск пересечений с ребрами графа. Это делает подход не менее эффективным, чем иерархические методы ПП [1].

$P = \{P_i, \text{такие что } P_i \in S \text{ и существуют } j, k \text{ такие, что } P_i \in R_j, R_j \in A_k\}$  (6).

6. Если найдена всего 1 ТП, то идет поиск сегментов НГ, перпендикуляр к которым из начальной или конечной точки имеет длину  $< \epsilon$ . Из числа найденных сегментов выбирается тот, длина перпендикуляра к которому меньше других, и к множеству  $P$  добавляется  $P1$  – ТП с перпендикуляром, образуя расширенное множество ТП  $P'$ :

$P' = P \cup P1$  (7).

7. Рассчитывается сортированное по удаленности от начальной точки пути множество ТП  $P_s$ .

$P_s = \{P_i, \text{такие что } D(P_i, P_n) < D(P_{i+1}, P_n) \text{ при } i \in 1..Np - 1\}$  (8),

где  $D(A, B)$  – расстояние от точки  $A$  до точки  $B$ ,  $Np$  – количество элементов множества  $P'$ .

8. Множество ТП разбивается на несколько частей  $P_{sk}$ , в каждой из которых идут подряд точки одного и того же графа.

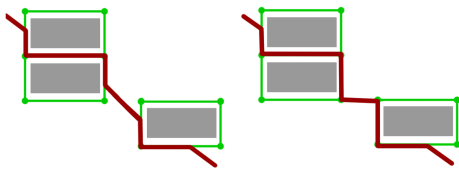
$P_{sk} = \{P_i, \text{такие что } P_i \in P_s, \text{ а также } P_i \in A_k, P_{i+1} \in A_k \text{ при } i \in 1..Np - 1\}$  (9).

9. Внутри получившихся частей списка ПП ведется отдельно, а крайние точки соседних частей

соединяются прямым отрезком. При ПП внутри одного НГ рассматриваются варианты:

- 1 ТП — путь касается НГ и поиск оптимального пути не требуется.
- 2 и более ТП — выполняется ПП по крайним ТП по критерию минимальной суммарной длины ребер. Ребра учитываются в соответствии с признаком проходимости  $T_A$ .

10. Компоновка полученных отрезков путей в единый путь. Объединение производится за счет того, что берутся найденные пути для каждого графа и крайние точки этих путей соединяются прямой линией.

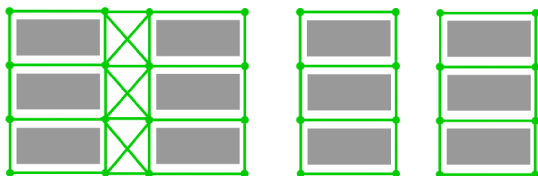


**Рис. 5:** ПП в модифицированном (слева) и базовом методах (справа) НГ. Серые прямоугольники - объекты, зеленые линии – НГ объектов, красные линии - найденные пути.

Описанный алгоритм, по крайней мере, не уступает в адекватности найденного пути базовому подходу, а в ряде случаев и превосходит его. Например, в усовершенствованном методе (Рис.5) путь по пространству свободному от препятствий зависит от координат начальной и конечной точек и проходит более естественно, в отличие от базового метода, где при относительно небольшом смещении начальной и конечной точек, путь проходит по одному и тому же ребру. Таким образом, расширенный алгоритм свободен от недостатков № 2, упомянутых в п.2.

Модифицированный алгоритм также позволяет реализовать ряд возможностей для сокращения объема ручной работы при задании НГ, что избавляет новый подход от недостатка метода НГ № 1:

1. Т.к. для каждого объекта используется свой НГ, пустое, гарантированно проходимое пространство не требует заполнения путями графа (Рис.6).
2. Для повторяющихся объектов НГ может быть скопирован автоматически.
3. Для обходных путей простой геометрической формы может быть использована автогенерация пути на основе данных о bounding box объекта и его положении в пространстве.



**Рис. 6:** НГ в базовом (слева) и модифицированном (справа) методах

## 4. УЧЕТ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

Динамические объекты учитываются также как статические, заданные переносом (позицией) и углом поворота на момент начала ПП.

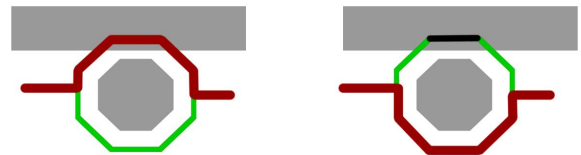
Рассмотрим приемы, которые позволяют учитывать динамические объекты в рамках общего алгоритма ПП.

### 4.1 Столкновение в момент следования по пути

Так как динамические объекты могут изменить свою позицию и угол поворота, то необходим механизм разрешения коллизии в момент следования по пути. В таких случаях предлагается находить путь заново из точки столкновения и, таким образом, учитывать новое положение динамического объекта.

### 4.2 Изменение состояния проходимости ребра

Ребро НГ, по которому идет найденный путь, может полностью оказаться в непроходимом пространстве. Чтобы учесть этот случай, в момент коллизии ребро отмечается как непроходимое с помощью признака проходимости  $T_A$  и при новом ПП не учитывается (Рис.7). Когда у динамического объекта изменяется позиция или угол поворота, все его ребра отмечаются как проходимые.



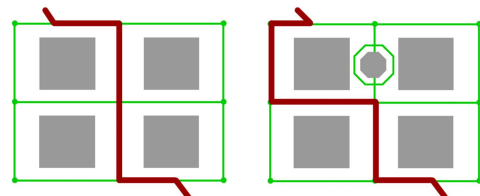
**Рис. 7:** Найденный путь обхода динамического объекта при первой попытке ПП (слева) и второй попытке (справа).

Черным цветом выделено ребро, отмеченное как непроходимое после первой попытки.

### 4.3 Учет динамических объектов, расположенных на ребре НГ статического объекта

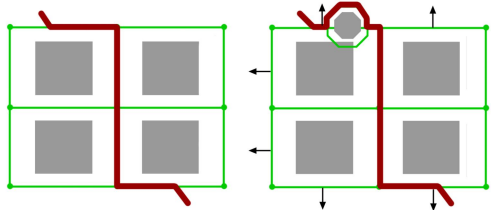
Для каждого ребра статического НГ задается контекстная информация  $T_D$  – информация о проходимости пространства вокруг ребра: ребро можно обойти слева, справа, слева и справа, нельзя обойти совсем. Эта информация используется для учета динамических объектов, расположенных на ребре НГ статического объекта. Возможны 2 случая:

- Ребро нельзя обойти. Оно помечается как непроходимое (Рис.8).



**Рис. 8:** ПП с динамическим объектом на ребре без заданного признака проходимости статического НГ (справа) и без него (слева).

- Ребро можно обойти. Тогда в п.10 алгоритма при сборке пути участок пути по рассматриваемому ребру между первой и последней ТП с динамическим объектом заменяется обходным путем динамического объекта. Направление обхода соответствует информации о проходимости (Рис.9).



**Рис. 9:** ПП с динамическим объектом на ребре с заданным признаком проходимости статического НГ (справа) и без него (слева).

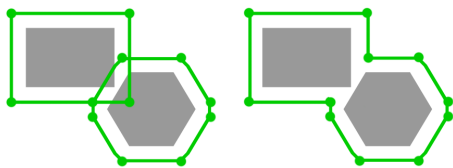
Пункты 4.2 и 4.3 позволяют расширенному методу НГ избавиться от третьего недостатка базового метода (п.2).

#### 4.4 Слияние НГ

Так как динамические объекты могут занимать произвольное положение в пространстве, то возможна ситуация, когда их НГ пересекаются. Может оказаться, что ребро одного НГ динамического объекта проходит через другой динамический объект, и это ребро входит в найденный путь, который окажется некорректным. Чтобы избежать такой ситуации, необходимо слияние графов, такое, что все ребра объединенного графа окажутся проходимыми (Рис.10).

Алгоритм слияния:

1. Поиск ТП графов.
2. Ребра, которым принадлежат ТП, делятся ТП на части. Эти части рассматриваются как новые ребра графа вместо исходного.
3. Проверяем все ребра и отбрасываем те, которые находятся внутри одного из исходных графов.



**Рис. 10:** НГ динамических объектов до слияния (слева) и после слияния (справа).

#### 5. СИНХРОНИЗАЦИЯ

Часто требуется показать перемещение персонажа по пути не только на своем компьютере, но и на компьютерах других пользователей. При этом учет динамических объектов не должен усложнить синхронизацию.

Путь считается только для своего персонажа. Для других пользователей пересылается набор контрольных точек пути. Это существенно сокращает объем расчетов. В худшем случае п.4.1, путь будет скорректирован и послан заново, как и в ситуации смены пользователем направления пути.

#### 6. ВЫВОДЫ

Чтобы получить представление об эффективности предложенного подхода, требуется оценить сложность учета динамических объектов. У п.1 две части. У первой части сложность  $O(n)$

Шаг	Описание	Сложность
1.	Поиск динамических объектов, у которых изменился поворот или позиция	$O(N)$ , где $N$ – число динамических объектов
2.	Пересчет координат вершин найденных динамических объектов	$O(\sum_{i=1}^k len(V_i))$ , где $k$ – число найденных динамических объектов, $len$ – количество элементов множества
3.	Поиск пересекающихся динамических объектов	$O(k*N)$

**Табл. 1:** Оценка сложности учета динамических объектов

Исходя из оценки сложности (Табл.1), можно сделать вывод, что алгоритм подходит для применения в реальном времени. При этом расширенный алгоритм свободен от 3 из 4 упомянутых в п.2. недостатков базового метода, что делает его более перспективным.

На основе описанного алгоритма была разработана программная система ПП в 3D-пространстве. Система была успешно апробирована в образовательной 3D-среде «Виртуальная Академия» [7].

#### 7. REFERENCES

- [1] A. Botea, M. Muller, and J. Schaeffer, "Near Optimal Hierarchical Path-finding," *Journal of Game Development*, vol. 1, issue 1, 2004.
- [2] Xiao Cui, Hao Shi, "A\*-based Pathfinding in Modern Computer Games," *IJCSNS International Journal of Computer Science and Network Security*, vol.11, no.1, January 2011.
- [3] Mika M., Charla C. Simple, "Cheap Pathfinding," *AI Game Programming Wisdom*, 2002.
- [4] John C. O'Neill, "Efficient Navigation Mesh Implementation," *Journal of Game Development*, vol. 1, no. 1, pp. 71-90, 2004.
- [5] Paul Tozour, "Fixing Pathfinding Once and For All", 2008. <http://www.ai-blog.net/archives/000152.html>.
- [6] Peter Yap, "Grid-Based Path-Finding," *AI '02 Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*.
- [7] <http://vacademia.com>.
- [8] <http://www.youtube.com/watch?v=NMunrmtG00o>.