

Хамитов М.А. Визуализация маршрута на Google картах с использованием промежуточных точек и алгоритма поиска оптимального пути // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №7 (июль). – АРТ 425-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.02

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

**ВИЗУАЛИЗАЦИЯ МАРШРУТА НА GOOGLE КАРТАХ С
ИСПОЛЬЗОВАНИЕМ ПРОМЕЖУТОЧНЫХ ТОЧЕК И
АЛГОРИТМА ПОИСКА ОПТИМАЛЬНОГО ПУТИ**

Аннотация: В статье рассмотрен метод визуализации маршрута на Google картах с использованием промежуточных точек и алгоритма поиска оптимального пути, а также реализация данного метода.

Ключевые слова: маршрут, оптимальный путь, алгоритм, координаты, местоположение, ГИС, Google карты, Google Fusion Tables.

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

VISUALIZATION OF THE ROUTE ON GOOGLE MAPS USING INTERMEDIATE POINTS AND THE ALGORITHM FOR FINDING THE OPTIMAL PATH

Abstract: In the article the method of visualization of a route on Google maps with use of intermediate points and algorithm of search of an optimal path, and implementation of this method.

Keywords: route, optimal path, algorithm, coordinates, location, GIS, Google Maps, Google Fusion Tables.

Зачастую, при разработке собственных карт, таких как технический план здания, карта небольшой местности и тому подобное, появляется проблема в пространственном ориентировании среди множества объектов на карте. Таким образом появляется проблема в поиске кратчайшего пути между двумя объектами, к примеру, из одной комнаты здания в другую. При ликвидации этой проблемы наиболее оптимальным решением является размещение промежуточных точек на карте, для дальнейшего построения маршрута по этим точкам.

Промежуточные точки

Входными данными разрабатываемой функции являются начальный и конечный идентификатор точек маршрута, а сами промежуточные точки на карте должны содержать информацию о соседних точках (список идентификаторов точек). Пример данных промежуточных точек представлен на таблице 1, где ID – идентификатор точки, NEARID – идентификатор соседних точек.

ID	NEARID
0	1
1	2,5,4
2	1,4,5
3	4
4	2,3,5
5	1,2,4,7
...	...

Таблица 1 – Данные промежуточных точек

Исходя из таблицы 1, построение маршрута от точки с ID «0» к точке с ID «3» напрямую невозможен. Для этого можно построить маршрут через точки с ID «1», «2», «4», либо «1», «4», либо «1», «5», «4» и т.д. Таким образом проявляется проблема в поиске оптимального пути. В нашем случае оптимальный путь содержит меньше промежуточных точек. Для организации данного поиска необходимо разработать алгоритм поиска оптимального пути.

Алгоритм поиска оптимального пути

Суть алгоритма заключается в обходе каждой маршрутной точки на карте, начиная с заданной начальной, и параллельном создании списка всех возможных путей до тех пор, пока не будет достигнута заданная конечная точка.

Для начала необходимо задать массив маршрутных точек, начальную и конечную точку маршрута, указать идентификатор текущей проверяемой точки и указать, что конечная точка еще не достигнута.

```
1 function pathFinding(pathData, SI, FI) {
2     var startIndex = SI;
3     var finIndex = FI;
4     var currentID = startIndex;
5     var reach = false;
6     ...
7 }
```

Листинг 1 – Функция поиска оптимального пути

После этого устанавливается цикл с условием: «Повторять до тех пор, пока конечная точка не достигнута». Внутри цикла производится создание списка путей, при этом проверяется, был ли проведен поиск по этой точке ранее, чтобы не создавать список путей повторно. Путь создается для каждой соседней точки от текущей проверяемой.

Затем текущая проверяемая точка помечается как проверенная (задается значение «-1»), и заменяется идентификатором следующей точки, которая является соседней.

Также проверяется, не является ли эта точка конечной. Если является, то цикл завершается.

```
1 function pathFinding(pathData, SI, FI) {
2     ...
3     while (reach === false) {
4         var near = [];
5         near = pathData[currentID].near.split(',');
6
7         near.forEach(function (id) {
8             if (paths[id] !== -1)
9                 paths[id] = paths[currentID] + ',' + id;
10        });
11
12        paths[currentID] = -1;
13
14        for (var i = 0; i < paths.length; i++) {
15            if (paths[i] !== undefined && paths[i] !== -1) {
16                currentID = i.toString();
17                break;
18            }
19        }
20    }
21 }
```

```
22     }
23
24     if (paths[finIndex] !== undefined)
25         reach = true;
26     }
...     ...
    }
```

Листинг 2 – Функция поиска оптимального пути

В конечном итоге получаем список с наименьшим количеством промежуточных точек.

Визуализация маршрута на Google картах

Для визуализации маршрута на Google картах можно использовать сервис Google Fusion Tables. Таким образом данные о промежуточных точках будут включать информацию о местоположении на языке KML. Получить координаты из KML кода можно при помощи XML парсера, а из полученных координат нарисовать линию на карте и расставить маркеры начальной и конечной точек.

```
1  function pathFinding(pathData, SI, FI) {
...     ...
27     var pathIds = paths[parseInt(finId)].split(',');
28     var pathCoordinates = [];
29
30     pathIds.forEach(function (pid) {
31         var parser = new DOMParser();
32         var kml = parser.parseFromString(pathData[pathData.map(function (e) {
33             return e.id;
34         }).indexOf(parseInt(pid))].geom, "text/xml");
35         var latLngs =
kml.getElementsByTagName("coordinates")[0].childNodes[0].nodeValue.split('
');
36         pathCoordinates.push({
37             lat: parseFloat(latLngs[0].split(',')[1]),
38             lng: parseFloat(latLngs[0].split(',')[0])
39         });
40     });
41
42     var markerA = new google.maps.Marker({
43         position: pathCoordinates[0],
```

```
44     map: map,  
45     label: 'A'  
46   });  
47  
48   var markerB = new google.maps.Marker({  
49     position: pathCoordinates[pathCoordinates.length - 1],  
50     map: map,  
51     label: 'B'  
52   });  
53  
54   var pathPolyline = new google.maps.Polyline({  
55     path: pathCoordinates,  
56     geodesic: true,  
57     strokeColor: '#ff3fa1',  
58     strokeOpacity: 1.0,  
59     strokeWeight: 3,  
60     map: map  
61   });  
62 }
```

Листинг 3 – Функция поиска оптимального пути

Результат работы функции на пользовательской Google карте представлен на рисунке 1.

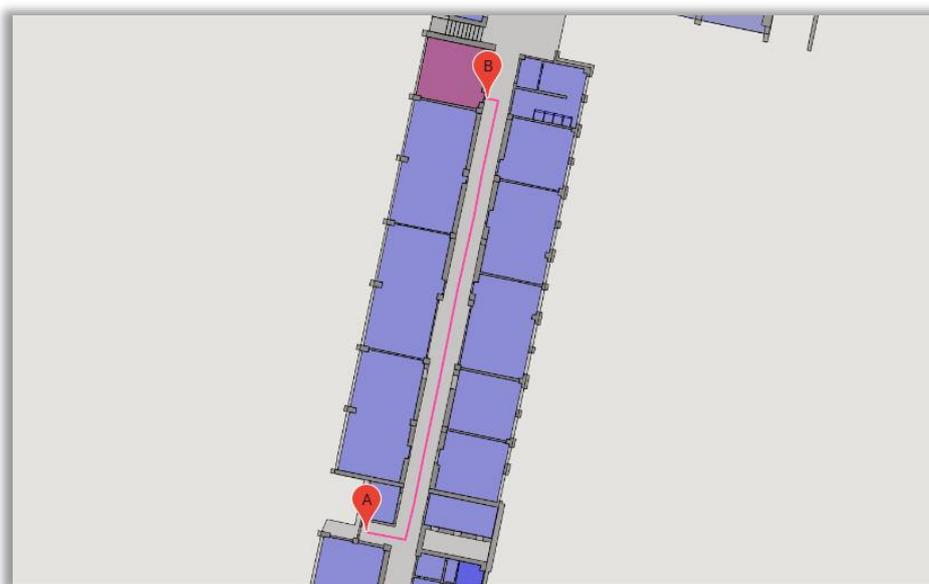


Рисунок 1 – Визуализация маршрута с использованием промежуточных точек

Подводя итог, можно сказать, что данный подход к визуализации маршрута на Google карте с использованием промежуточных точек является в достаточной степени быстрым и точным при определении оптимального маршрута. Главным достоинством данной реализации является простота и скорость организации данных промежуточных точек, так как они включают лишь идентификаторы самой точки и соседних точек.

Список использованной литературы:

1. Google Maps JavaScript API // Google: документация программного интерфейса. 2018. URL: <https://developers.google.com/maps/documentation/javascript/tutorial>, (дата обращения 04.07.2018)
2. Google Fusion Tables // Google Fusion Tables: центр помощи. 2018. URL: <https://fusiontables.google.com/>, (дата обращения 04.07.2018)

Дата поступления в редакцию: 02.07.2018 г.

Опубликовано: 06.07.2018 г.

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

© Хамитов М.А., 2018