

Implementação do algoritmo A* pathfinding

Adriano Costa¹, Steffany Santos Hadlich¹

¹Departamento de Computação – Universidade Estadual do Centro - Oeste

adriano.kosta@yahoo.com.br, tefani.hadlich@gmail.com

1. Resumo

Este documento tem como objetivo a demonstração de uma implementação do algoritmo A* Pathfinding, para fins de aprendizado do mesmo. A aplicação consiste em uma apresentação aonde o algoritmo deve calcular o menor caminho até o objetivo, desviando de possíveis obstáculos.

2. Introdução

O algoritmo A* *pathfinding* consiste no cálculo da avaliação global do determinado nó até o objetivo, é dado pela equação : $f = g + h$. Onde a distância f é dada pela soma do o custo de movimento do nó atual ao seu vizinho, determinada por g e a distância de seu vizinho até o estado final, determinada por h .

Os nós são analisados através de duas listas, sendo elas a de nós abertos, que contém os nós que ainda precisam ser verificados, e a lista de fechados, que contém os nós que já foram verificados.

3. A apresentação

A apresentação do algoritmo ao usuário é feita através de um *grid*¹, onde cada quadrado representa um "pedaço" do caminho até o final, que irá demonstrar através de cores qual é o menor caminho, os nós que foram fechados, e os nós que ainda estão na lista de abertos.

A imagem 1 demonstra após aplicado o algoritmo, onde o quadrado superior esquerdo é a origem e o quadrado inferior a direita, representa o destino. A cor azul representa o melhor caminho encontrado pelo algoritmo, o vermelho mostra os nós que foram consultados, o verde os nós que ainda deveriam ser consultados caso selecionados, o preto por sua vez demonstra alguns obstáculos que o algoritmo poderia encontrar e por ultimo os pontos amarelos indicam os estados inicial e final que o algoritmo deve ligar.

4. Implementação

A implementação foi feita com base no pseudocódigo disponibilizado no site [Wikipedia 2017]. Foi utilizada a linguagem *javascript* para a construção da lógica, e *html* para apresentação da interface ao usuário. Para a construção da interface foi utilizado a biblioteca *p5**, feita para desenhos gráficos. Suas funções permitem que o usuário faça de uma forma simples desenhos utilizando as funções *draw* e *setup*, que geram um *loop* exibindo passo a passo a execução do código este sendo finalizado quando o critério de parada for atendido, ou seja, quando o nó final entra para lista de fechados e a função *noLoop()* interrompe o laço ou quando a lista de abertos se encontra vazia, o que indica

¹Se trata de uma grade de células que representa o espaço, estruturado como uma matriz

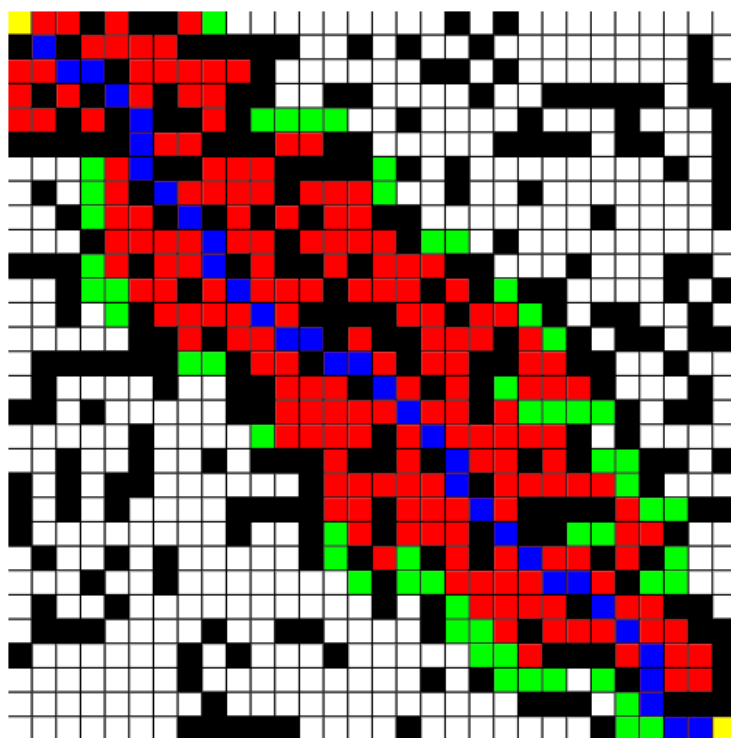


Figura 1. Apresentação após execução do algoritmo.

que o problema não apresenta uma solução. A aplicação consiste em uma matriz como demonstrado na figura 1, onde cada posição da matriz representa um nó do caminho a ser percorrido.

Para o cálculo da distância do nó até o estado final representado por h foi utilizado a Distância Manhattan, que consiste no calculo da distância utilizando a posição horizontal e vertical. A fórmula geral da Distância Manhattan é dada por $|x_2 - x_1| + |y_2 - y_1|$, aonde no problema aplicado x_1 representa a primeira coordenada do nó a ser verificado, e x_2 a primeira coordenada do nó que representa o estado final, e a mesma coisa acontece na segunda parte da equação, y representa a segunda coordenada do nó e 1 e 2 respectivamente, o nó atual a ser verificado e o estado final.

Um exemplo prático é considerar que a distância ortogonal ou seja tanto no movimento horizontal ou vertical, a distância até o próximo nó é 10, e no movimento diagonal a distância é 14, seguindo essa lógica e adicionando esse valor á distância do nó atual até o nó final sem considerar os obstáculos, o valor do caminho mais curto é calculado.

5. Conclusão

O algoritmo *A* pathfinding* é interessante por prever o caminho a ser explorado levando em consideração não só a distância entre o nó e seu vizinho mas por estimar também a distância desse vizinho até o nó final, utilizando heurística, diferente por exemplo do algoritmo de *Dijkstra*.

A implementação do código nesse trabalho foi simples, apenas para encontrar o menor caminho de um ponto A para outro ponto B desviando de obstáculos no caminho, mas possibilidades de aplicação são amplas, podendo ser utilizada para mapeamento de

espaços, jogos de vídeo game, projetos para construções de ferrovias dentre outros.

O trabalho proporcionou a equipe a oportunidade de explorar novas linguagens para desenvolvimento, como o javascript e html além do estudo de formas de apresentação do funcionamento do código em uma interface, que foi obtido através da biblioteca p5* que une código e arte dentro de um mesmo domínio.

Referências

- Eranki, R. (2002). Pathfinding using a* (a-star). <http://web.mit.edu/eranki/www/tutorials/search/>. [Online; acessado em 29 Abril 2017].
- Johnson, J. (2017). p5.js. <https://p5js.org/reference/#/p5/draw>. [Online; acessado em 1 Maio 2017].
- NIST (2017). Manhattan distance. <https://xlinux.nist.gov/dads/HTML/manhattanDistance.html>. [Online; acessado em 1 Maio 2017].
- Shiffman, D. (2017). The Coding Train. <http://thecodingtrain.com>. [Online; acessado em 29 Abril 2017].
- Wikipedia (2017). A* search algorithm. https://en.wikipedia.org/wiki/A*_search_algorithm. [Online; acessado em 1 Maio 2017].