游戏编程算法与技巧

向量减法 B-A 由A指向B

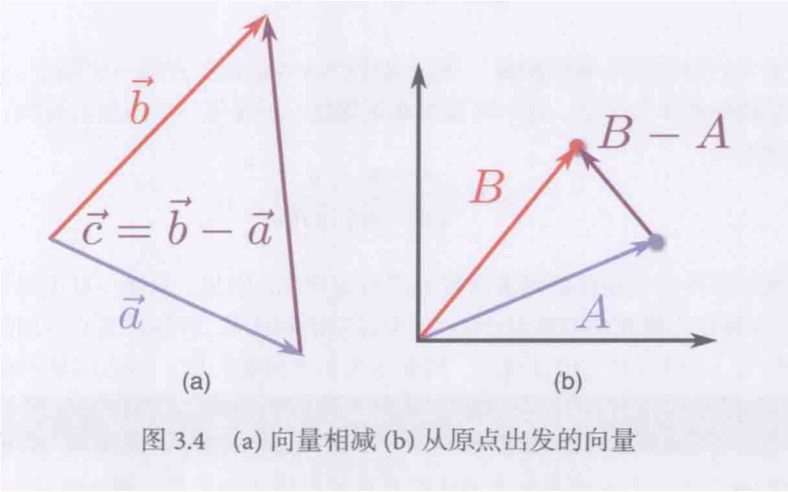


图 3.4 (a) 向量相减 (b) 从原点出发的向量

矩阵乘法:

$$C = A \times B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a \cdot e + b \cdot g & a \cdot f + b \cdot h \\ c \cdot e + d \cdot g & c \cdot f + d \cdot h \end{bmatrix}$$

$$\begin{bmatrix} a & b \end{bmatrix} \times \begin{bmatrix} c & d \\ e & f \end{bmatrix} = \begin{bmatrix} a \cdot c + b \cdot e & a \cdot d + b \cdot f \end{bmatrix}$$

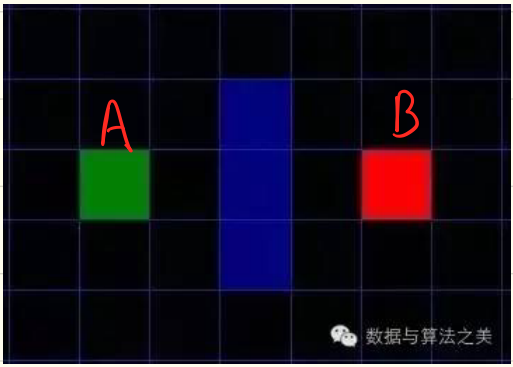
转置矩阵

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

齐次坐标 w是0 则为3D下向量

w是1 则为3D下点。

A*算法



从起点A开始，并把它加入到一个由方格组成的open list（开放列表）中。open list内现在只有一项。

open list里的格子是路径可能会沿途经过的，也有可能不经过。

open list是一个待检查的方格列表。

查看与起点A相邻的方格，把其中可走的方格也加入到openlist中。把起点A设为这些方格的父亲。

把A从open list中移除，加入到close list（封闭列表）中，close list中的每个方格都是现在不需要再关注的。

路径排序

计算出组成路径的方格的关键是下面这个等式：

$F=G+H$

G=从起点、A移动到指定方格的移动代价沿着到达该方格而生成的路径。

H=从指定的方格移动到终点B的估算成本。

路径是反复遍历open list，选择F值最小的方格。

A*算法相对广度优先搜索算法，除了考虑中间某个点同出发点的距离以外，还考虑这个点同目标点的距离。

S→M的距离 + M→E的距离。如果我们用符号表示的话，就可以写成：f(M) = g(M) + h(M)。