算法 1 场景整体规划

输入： 空白场地 $ground$，T 个虚拟关卡信息 $virlevel_{1...T}$(与前一关距离 $dist$，偏转角度 $deflex$，形状 $shape$)

输出： T 个关卡的实际情况 $realevel_{1...T}$(实际位置 $pos(x,y)$,实际形状 $shape$)

将场地 $ground$ 划分为 $n \times m$ 个边长为 0.5 米的方格
$flag_{1...T}$, $realevel.shape \leftarrow$ `MakeShape`$(ground, virlevel)$ // 预处理每个关卡可达方格最多的自旋角度，见算法 2
$map_{1...T} \leftarrow$ `MakeMap`$(virlevel, flag)$ // 动态规划预处理每个关卡从每个方格到出口的最小偏转偏差，见算法 3
cnt_{max}, $\theta_{min} \leftarrow 0, \infty$
for $bits \leftarrow 0 \ldots (1 << T) - 1$ **do**
    $pos_{1...T}$, cnt_{all}, $\theta_{all} \leftarrow$ `MakePos`$(virlevel, realevel, flag, map, bits)$ // $bits$ 方案下的贪心策略，见算法 4
    if $cnt_{max} < cnt_{all}$ `or` $(cnt_{max} == cnt_{all}$ `and` $\theta_{min} > theta_{all})$ **then**
        $realevel_{1...T}.pos$, cnt_{max}, $theta_{min} \leftarrow pos_{1...T}$, cnt_{all}, θ_{all}
    end if
end for

算法 2 `MakeShape` $(ground, virlevel)$

输入： 空白场地 $ground$，T 个虚拟关卡信息 $virlevel_{1...T}$

输出： T 个关卡在每个方格上的可达性 $flag_{1...T}$，最优自旋角度下的新形状 $realevel_{1...T}.shape$

$\alpha \leftarrow 0$ // 前面若干关卡累计自旋角度和
for $k \leftarrow 1 \ldots T$ **do**
    $virlevel_k.deflex$, $cntmax$, $\beta \leftarrow virlevel_k.deflex + \alpha$, 0, 0
    for $\gamma \leftarrow 0 \ldots 360^\circ$ **do**
        $cnt \leftarrow 0$
        for i, $j \leftarrow 1 \ldots n$, $1 \ldots m$ **do**
            $tempshape \leftarrow shape$ 逆时针旋转$\alpha + \gamma$
            if $tempshape(i,j) \subseteq ground$ **then** // $tempshape(i,j)$表示$tempshape$在i行j列处时包含的区域，下同
                $cnt \leftarrow cnt + 1$
            end if
        end for
        if $cntmax < cnt$ **then** $cntmax$, $\beta \leftarrow cnt$, γ
        end if
    end for
    for i, $j \leftarrow 1 \ldots n$, $1 \ldots m$ **do**
        $realevel_k.shape \leftarrow shape$ 逆时针旋转$\alpha + \beta$
        if $realevel_k.shape(i,j) \subseteq ground$ **then** $flag_k[i,j] \leftarrow$ `true`
        else $flag_k[i,j] \leftarrow$ `false`
        end if
    end for
    $\alpha \leftarrow \alpha + \beta$
end for
for $k \leftarrow 1 \ldots T - 1$ **do**
    for i, $j \leftarrow 1 \ldots n$, $1 \ldots m$ **do**
        if $flag_k[i,j]$ `is false` **then** `continue`
        end if
        $flag_k[i,j] \leftarrow$ `false` // i行j列放置当前关卡后，若下一关没有空地，则也视为不可达
        for i', $j' \leftarrow 1 \ldots n$, $1 \ldots m$ **do**
            if $flag_{k+1}[i',j']$ `isn't false and` $realevel_k.shape(i,j) \cap realevel_{k+1}.shape(i',j') = \phi$ **then**
                $flag_k[i,j] \leftarrow$ `true`
            end if
        end for
    end for
end for

算法 3 `MakeMap` $(virlevel, flag)$

输入： T 个虚拟关卡信息 $virlevel_{1...T}$, 预处理结果 $flag_{1...T}$

输出： 每个关卡到从每个方格到出口的最小偏转偏差 $map_{1...T}$

$map_{1...T} \leftarrow [\infty, \ldots, \infty][\infty, \ldots, \infty]$
for $k \leftarrow T \ldots 1$ **do**
  for $i,\ j,\ i',\ j' \leftarrow 1 \ldots n,\ 1 \ldots m,\ 1 \ldots n,\ 1 \ldots m$ **do**
    if $flag_k[i,j]$ `is false or` $flag_{k+1}[i',j']$ `is false then continue`
    else if $realevel_k.shape(i,j) \cap\ realevel_{k+1}.shape(i',j') \neq \phi$ **then** `continue`
    else if (i,j)所处位置到(i',j')所处位置距离与$virlevel_{k+1}.dist$差距过大 **then** `continue`
    end if
    $\Delta\theta \leftarrow$ 从(i,j)到(i',j')角度与$virlevel_{k+1}.deflex$之差的绝对值
    $map_k[i,j] = \min\left\{map_k[i,j],\ map_{k+1}[i',j'] + \Delta\theta\right\}$
  end for
end for

算法 4 `MakePos` $(virlevel, realevel, flag, map, bits)$

输入： T 个虚拟关卡信息 $virlevel_{1...T}$，实际情况 $realevel_{1...T}$，预处理结果 $flag_{1...T}$，$map_{1...T}$，贪心方案 $bits$

输出： $bits$ 方案下的贪心策略结果 $pos_{1...T}$，覆盖方块数 cnt_{all}，偏转角度总偏差 θ_{all}

$mask[1,1], \ldots, mask[n,m] \leftarrow$ `false`
$cnt_{all},\ \theta_{all} \leftarrow 0,\ 0$
for $k \leftarrow 1 \ldots T$ **do** // 从头到尾逐个规划，每个关卡选坪效最大或偏转角度偏差最小的位置
  $cnt_{tempmax},\ \theta_{tempmin} \leftarrow 0,\ \infty$
  for $i,\ j \leftarrow 1 \ldots n,\ 1 \ldots m$ **do**
    if $flag_k[i,j]$ `is false then continue`
    else if $realevel_{k-1}.shape(pos_{k-1}) \cap\ realevel_k.shape(i,j) \neq \phi$ **then** `continue`
    else if (i,j)所处位置到pos_{k-1}距离与$virlevel_k.dist$差距过大 **then** `continue`
    end if
    $cnt \leftarrow realevel_k.shape(i,j)$中满足$mask[x,y]$为 `false` 的方格$(x,y)$数量
    $\theta \leftarrow \theta_{all} + map_k[i,j] +$ (i,j)相对pos_{k-1}的偏转角度与$virlevel_k.deflex$之差的绝对值
    if $bits >> (k-1)$ `is 1 then` // $bits$为1优先考虑增加坪效
      if $cnt_{tempmax} < cnt$ `or` $(cnt_{tempmax} == cnt$ `and` $\theta_{tempmin} > \theta)$ **then**
        $p,\ cnt_{tempmax},\ \theta_{tempmin} \leftarrow (i,j),\ cnt,\ \theta$
      end if
    else // $bits$为0优先考虑减少偏转偏差
      if $\theta_{tempmin} > \theta$ `or` $(\theta_{tempmin} == \theta$ `and` $cnt_{tempmax} < cnt)$ **then**
        $p,\ cnt_{tempmax},\ \theta_{tempmin} \leftarrow (i,j),\ cnt,\ \theta$
      end if
    end if
  end for
  $pos_k \leftarrow p$所处位置
  $\theta_{all} \leftarrow \theta_{all} +$ 从pos_{k-1}到pos_k角度与$virlevel_k.deflex$之差的绝对值
  for each $(i,j) \in realevel_k.shape(p)$ **do** $mask[i,j] \leftarrow$ `true`
  end for
end for
for $i,\ j \leftarrow 1 \ldots n,\ 1 \ldots m$ **do**
  if $mask[i,j]$ `is true then` $cnt_{all} \leftarrow cnt_{all} + 1$
  end if
end for