

程序逻辑--使用算法：贪心算法（左右方向选择时）【1】：

我们规划了四个区域作为图像抓取区域，分别是左方视图、右方视图、前方视图和障碍物视图。程序启动时，计算机只会识别和计算这些区域的色块像素（黑色/蓝色），并将其存储在实例中，以便稍后进行比较

- 左方视图：左上坐标 y 高于人物，右下坐标 y 低于人物。
- 右方视图：要求和面积与左方视图相同
- 前方视图：宽度 \leq 右方视图右下角横坐标 - 左方视图左上角横坐标
- 障碍物视图：宽度等于人物宽度，高度根据计算机运行速度和操作延迟而定

通过比较左右方向的 "黑色 "计数，优先选择 "黑色 "计数最少的方向。/通过比较左右方向的 "蓝色 "计数，优先选择 "蓝色 "计数最高的方向。【1】实时评估整个过程，修正规则，支持调试输出，便于分析。确保在动态环境中做出最佳决策。当整个下部向前移动时，避开障碍物。当识别到下部方向（下方视图以及障碍物视图）有黑色时/当蓝色识别率低于正常水平时，只选择左或右方向。当左右方向差异较小时，选择 "向下 "方向，忽略误差。

性能：

通过识别蓝色选择路径：平均 1000m

通过识别黑色选择路径：平均 4000m

问题偏向：

在大部分情况都能完成躲避，遇到无解（即来不及躲避或运算速度跟不上操作速度容易发生碰撞），容易发生擦边现象

优点缺点：

实际运行时避障效果最佳，拥有紧急避障逻辑和误差容错。运算判断速度最快，但容易陷入局部最优解

避障区域分为两层分别是宽度更大单更短的前方视图和更窄更长障碍物视图，减少由于靠近而产生的躲避空间不足的问题

计算判断到操作反应的时间仍然过长，且必须要在速度较低的情况运行

算法方式相对单一，容易由于随机性产生的噪音造成判断失误。

当前问题以及改进意见：

在不同电脑上参数需要手动调整，手动调整的参数包括：人物定位，蓝色正常水平数值，深度搜索计算深度，避障启动的距离，左右误差容忍范围等。

程序没有考虑游戏奖励机制，包括加速，跳跃等，黑色识别的方式没有考虑红色区域的减速机制以及非惩罚性障碍的影响。蓝色识别的方式将奖励也视为障碍物。

程序由于起点处图像的不对称性，以及不同人物的选择，容易影响方向的识别，造成在起点处发生碰撞现象。

游戏机制相对复杂，且噪音相对较多，障碍物随机性较强，无法单一算法操作不够精细，操作反应相对较长，无法针对所有的障碍物完成避障，造成碰撞。