

AI of WeChat's Jump

E-Neo

January 6, 2018

Outline

遇到的问题与解决方案

背景色与目标接近边缘检测错误

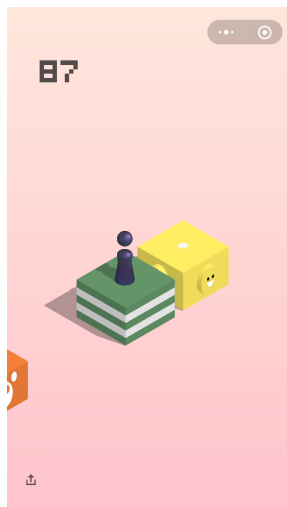


Figure: 游戏画面

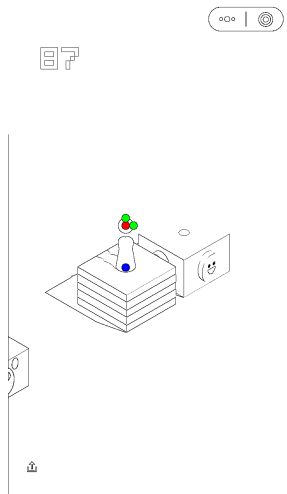


Figure: 处理后的图像

解决方案

修改 `process()` 函数：将：

```
img = img.convert('L').filter(ImageFilter.CONTOUR)
```

调整为：

```
img = img.filter(ImageFilter.CONTOUR).convert('L')
```

避免转化为灰度图后丢失有价值信息。

效果对比

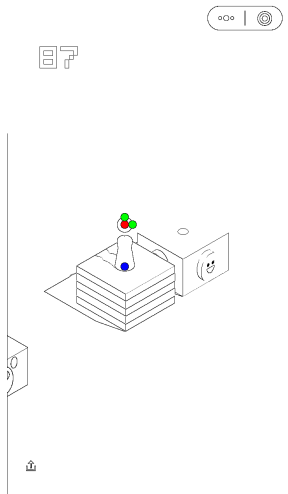


Figure: 修改前

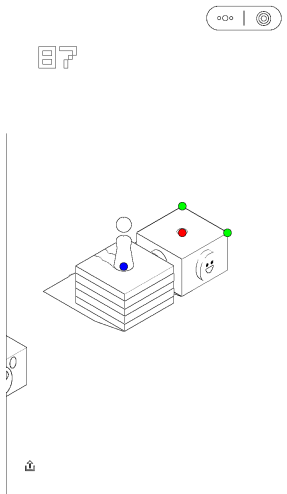


Figure: 修改后

将盒子侧棱误判为盒盖

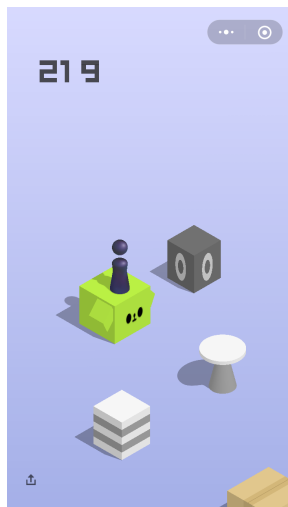


Figure: 游戏画面

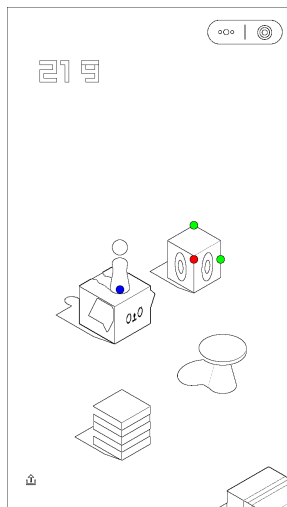


Figure: 处理后的图像

解决方案

find_d_right() 函数中：原先监测到目标最右端之后会沿边缘一直向下行走，导致将侧棱误判为盒盖。修改后：

```
else:
    while pixel[i, j] != 0:
        i = i - 1
    k = j
    while pixel[i, k] == 0:
        k = k + 1
    if k - j > 10:
        k = j + 2
    return i, (j + k) >> 1
```

效果对比

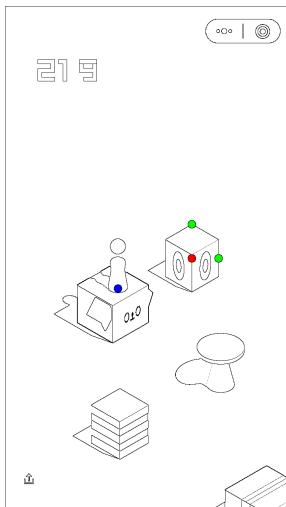


Figure: 修改前

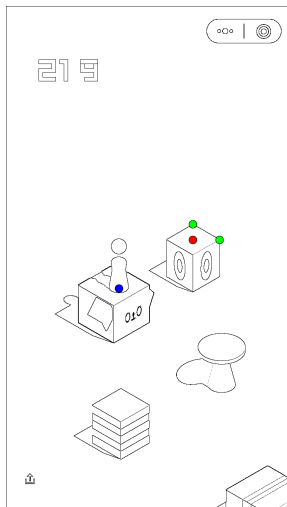


Figure: 修改后

目标不一定是最高点

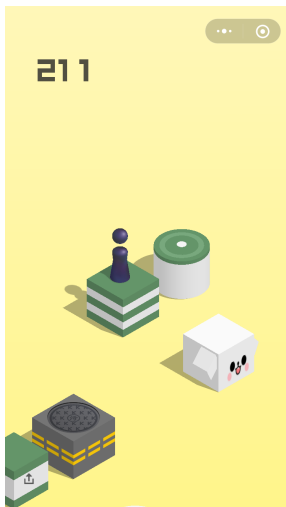


Figure: 游戏画面

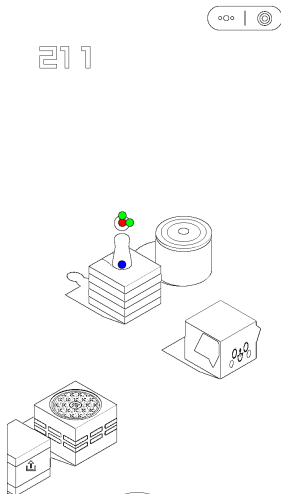


Figure: 处理后的图像

解决方案

- ▶ 原先寻找目标的方法是：从上往下扫描，找到第一个黑色的像素点即判定为目标物体的最高点。
- ▶ 将其修改为：在不包含 i 的区域内，从上往下扫描，找到第一个黑色的像素点即判定为目标物体的最高点。

效果对比

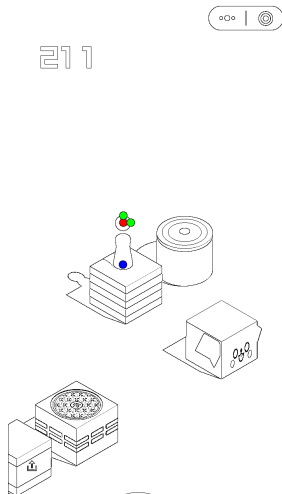


Figure: 修改前

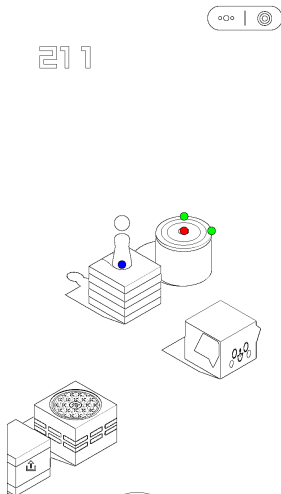


Figure: 修改后

游戏中其他标识与 i 颜色接近

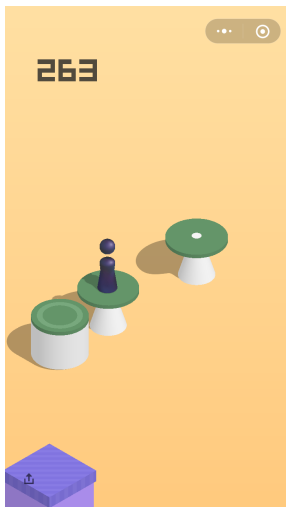


Figure: 游戏画面

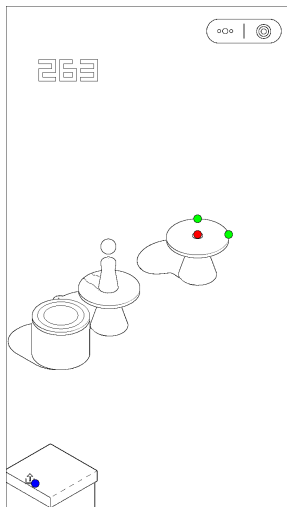


Figure: 处理后的图像

解决方案

重新限定 i 的检测范围，越过底部标识。

效果对比

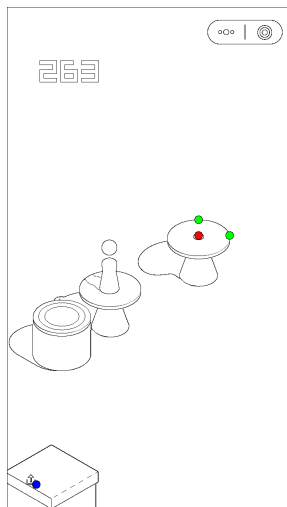


Figure: 修改前

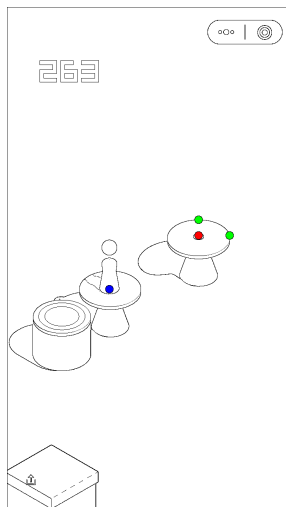


Figure: 修改后

End

Thank you for your attention!