## D4结束时的状态



## 屏幕刷新函数

```
co [lcd_refresh]
执行
        = (210)
    X
               中的每个项目 🕕
    对(
        map_list
    执行
             = ( 0
         C y
               中的每个项目 data
         对(
         执行
                                      区 位图纵坐标()
               实例 [ map_draw
                              位图横坐标
                                                   y
               索引
                    data
                  += - 30
             -= • 30
```

x方向移动函数

```
o boy_move_x 参数: x, y, step
执行
    使用全局变量 map_list
      ret
         = ( -1
                        实例
                             map_list
     taget_value
                   实例
                        索引
                             у
                   索引
                                 step
                         X
    🧿 如果 🚺 taget_value
                            0
                       执行
          实例
               map_list
                       赋值第
                                             项 值 4
                               X + *
                                       step
          索引
               y
               map_list
                       赋值第 x 项值 0
           实例
           索引
               у
                0
          ret
                  返回(ret)
```

## 函数调用:

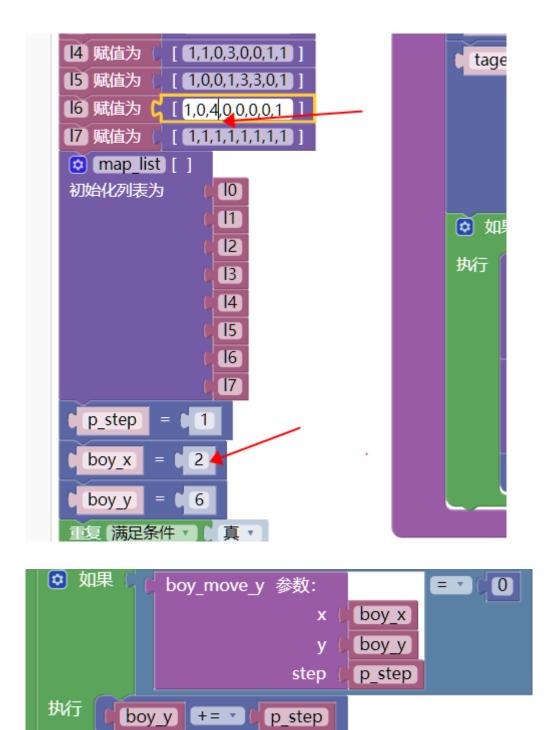
```
如果 boy_move_x 参数:
    x boy_x
    y boy_y
    step p_step

执行 boy_x += x p_step
```

y方向移动函数

```
o boy_move_y 参数: x, y, step
执行
    使用全局变量 map_list
     ret = 1 -1
                             map_list
     taget_value
                  实例
                        实例
                        索引
                               y + T step
                   索引
                       X
    💆 如果
             taget_value
                           0
    执行
          实例
               map_list
                               赋值第 x 项值 4
          索引
                 y + -
                        step
                       赋值第 🙀 🗴 项 值 関 0
               map_list
          实例
          索引
               У
              = [ 0
          ret
                  返回
                       ret
```

函数调用

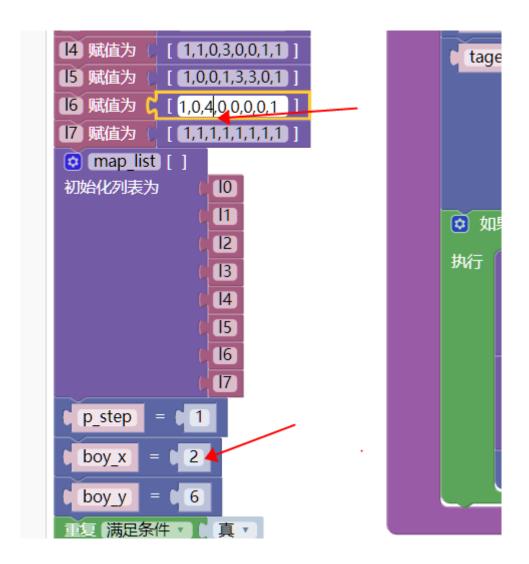


整合move\_x move\_y函数为move\_xy

```
🔯 (boy_move_xy) 参数: x, y, x_step, y_step
执行
    使用全局变量
               map_list
    ret = ( -1
     taget_value
                   实例
                         实例
                              map_list
                         索引
                               y + •
                                       y_step
                   索引
                          X + *
                                 x_step
    🧔 如果
             taget_value
                       = * (0
    执行
                                                         项值 4
                                 赋值第
           实例
                map_list
                                         x + x x_step
           索引
                 y + v y_step
                        赋值第 x 项值 0
                map_list
           实例
           索引(
               у
          ret = 0
                          返回 (ret)
```

### 函数调用

```
重复 满足条件 🔻 📋 📮
执行
    延时 毫秒 1000
    执行 lcd_refresh
    ♥ 如果
              boy move xy 参数:
                                           0
                            x boy_x
                               boy_y
                         x_step
                               p_step
                               0
                         y_step
    执行
          boy_x += *
                       p_step
```



## 图标制作

0. 找合适的图标

## https://www.iconfont.cn/

- 1. 用微信或其他工具截屏 截图大小长宽要一致
- 2. 修改图片大小到30x30像素,可在线完成

# https://www.gaitubao.com/#

- 3. 修改图标方向, 打开, 合适的旋转, 保存
- 4. 利用图标转换工具, 把图片转换为图标文件



使用方法:图片拖放到工具窗口,即可自动生成图标文件到图片所在目录。

5. 图标文件重命名

名字与Blockly中引用一致

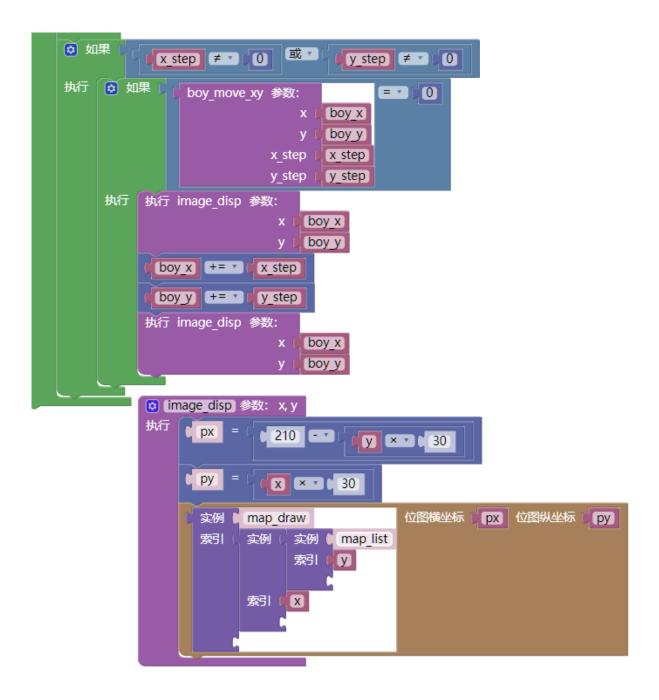
## 6. 下载使用

# 按键与移动结合

a. 按键部分

```
执行 lcd refresh
重复 满足条件 * 真 *
执行
    延时 室秒 ▼ (20)
    执行 key_scan
    i = 0
     x_step = ( 0
     y_step = 0
    对(key_state)中的每个项目(state)
    执行 🔯 如果 🛚
                state = 1 1
         执行
             key state 赋值第 (i) 项值 (0)
              ◎ 如果
                       i = v = 0
              执行
                   x step = ( 1
              否则如果
                       (i) = 17 (i) (i)
             执行 🔽
                   y step
              否则如果
                       () (37
                              2
             执行 x step = 1-1
              否则如果
                              3
              执行
                   y step
          () (#=17)
```

## b. 移动部分



## 目标位置移动

boy\_move\_xy中增加目标位置判断

#### 思路:

- 1. 记录目标位置: 把目标位置存入列表中
- 2. 当小人离开目标位置时,进行判断:遍历列表(target\_list),判断是否与当前小人所在 行列值相同
- 3. 如果当前位置是目标位置,那么显示星星,否则显示空白

```
使用全局变量
         map_draw
使用全局变量
         target_list
ret = ( -1
taget_value =
             实例
                  实例
                       map_list
                  索引
                        y + v y_step
             索引
                   x + x x_step
◎ 如果
                           或▼
          taget_value = 100
                                 taget_value = v (2)
执行
                                                项值 4
          map_list
                          赋值 第
     实例
                                  x + x x_step
     索引
           y + v y step
    tmp = (0)
    对 target_list 中的每个项目 target
    执行 🔯 如果
                                     实例 (target = x x
                                            实例
                                                 target = v (y)
                   索引 (0)
                                            索引 1
            (tmp) = (2)
     实例
          map_list
                 赋值第 X 项值 (tmp)
     索引
     ret = ( 0
                   返回 (ret
```

```
DOY
10 赋值为
            [ (0,0,1,1,1,1,0,0) ]
            [ (0,0,1 2,2 1,0,0 ]
11 赋值为(
            [ 0,1,1 0,2 1,1,0 ]
12 赋值为 [
            [ (0,1,0,0,3,2,1,0) ]
13 赋值为 [
[4] 赋值为 [
            [ 1,1,0,3,0,0,1,1 ]
            [ (1,0,0,1,3,3,0,1) ]
15 赋值为
            [ (1,0,4,0,0,0,0,1) ]
16 赋值为
[7] 赋值为 [ [1,1,1,1,1,1,1,1] ]
map list [ ]
                 10
初始化列表为
                  11
                  12
                  13
                  14
                  I5
                  16
                  17
🔯 (target_list) [ ]
                   [ 3,1 ]
                    [ 4,1 ]
                    [ 4,2 ]
                    [ [5,4]]
  p_step = (1)
 boy_x = (2)
```

函数拆分

```
🔯 (boy_move_xy) 参数: x, y, x_step, y_step
执行
     使用全局变量 (map list)
     ret = 1 -1
      taget_value
                     实例
                           实例
                                map list
                           索引
                                  y + v
                                          y_step
                     索引
                            X + *
                                     x step
     🔯 如果 📗
                                     或▼
                 taget_value = *
                                 0
                                             taget value = *
                                                             2
     执行
         执行 boy move 参数:
                              X
                              y
                              x step
                      x_step
                      y_step
                              y_step
           ret = 0
                            返回 ret
```

```
🔯 boy_move 参数: x, y, x_step, y_step
执行
    使用全局变量 | map_list
    使用全局变量 map_draw
    使用全局变量 target list
                                                 项值(4)
                           赋值 第
      实例(
          map list
                                   x + v x step
      索引(
          y + v y step
    tmp = (0
    对 (target_list) 中的每个项目 (target)
    执行
        🚺 如果
                                      且了
                        target = x x
                                                   target = v (y)
                   实例
                                              实例
                                                 索引 0
                                              索引
         执行
              tmp = 1 2
                  赋值第 🗽 项值
          map list
      实例
                                 tmp
      索引
          у
```

#### 程序自启动

需要自动执行的文件上右键,选择 "run default"

D6:

游戏设计主线

1. 屏幕显示

屏幕画图---》屏幕画地图 游戏坐标与屏幕坐标

2. 图标移动与限位

根据地图矩阵的数据是否为0判断能不能移动

- 3. 图标移动到星星位置,需要记录星星的位置,以便小人离开时还原
- 4. 推动箱子,箱子后面是空白或星星时可推动 怎么推
- 5. 胜负判定

### 原来星位置的值变成箱子的值

思考:

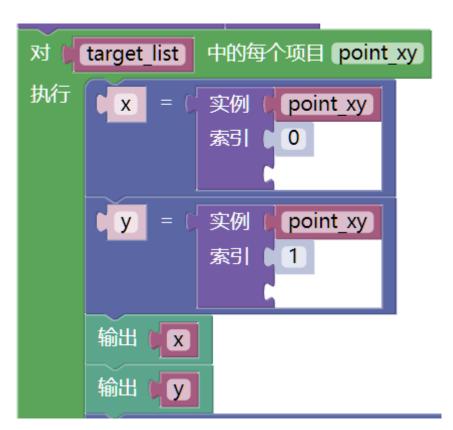
地图上没有星星是否说明游戏胜利?

### 挑战:

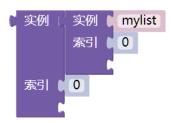
# 星星的坐标存储在target list中

1. 利用所用到的积木块,完成对target\_list的遍历,从中得到坐标值x,y。可以用打印的方法

```
target_list [ ]
初始化列表为
               [ 3,1 ]
               [ 4,1 ]
               [ 4,2 ]
               [ 5,3 ]
             中的每个项目(i)"(对target_list中的每个值执行)"
对
   0,1,2,3
执行
             "(获取target list中的每个元素的[0]处的值)"
     X
             " 获取target_list中的每个元素的[1]处的值 »
     輸出
         X
    输出(y
```



2. 然后,在map\_list中找到x,y坐标中的数据。也可以用打印方法验证如何利用索引嵌套获取二维矩阵中的数据



```
输出 (文) a = (实例 (map list 索引 (文) 索引 (文) 索引 (文) 索引 (文) 索引 (文) 索引 (文) 索引 (文)
```

3. 判断数据是否为3. 如果有任何1个不为3, 说明游戏没有胜利用打印的方法输出结果。

定义一个变量,用变量的值表示胜负

4. 附加功能

挑战任务: 练习与测试

记录小人移动步数,并在屏幕上进行显示

参考-把变量显示在屏幕上



5. 自动获取星星所在位置坐标的列表,如下表为手动完成的,请编写代码自动完



#### 思路:

成

- 1. 新建1个空的list: target\_list
- 2. 查找地图中每一个数据,在值为2的地方,记录当前的x,y值,把x,y组合为一个列表
  - 3. 把组合出的列表加入到target\_list

#### 参考:

创建空list,利用变量组合一个list,把1个list添加到另一个list