

dlc地图设计教程

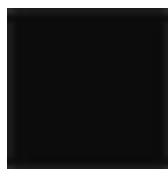
游戏设计：莫浩宇，张浩阳

游戏地图坐标及地图读入原理



游戏地图坐标及地图读入原理

0 675
24 675
48 675
72 675 1
96 675



仅用两个数字给定x, y坐标
可以在地图中导入一个标准方块(24*24)
若在后加上1, 此方块将只显示但没有任何效果
由于需要对应地图碰撞箱,
标准方块位置参数要求x坐标可以表示为: $24i+1$
否则可能出现边界不匹配现象

120 675
120 654 trap
120 631 trap
144 675



在坐标之后附加 trap 导入一个陷阱方块(21*21)
可选参数move+数字可以在角色到达陷阱方块正上方和正下方时触发方块移动, 数字代表移动速度, 正向下, 负向上 (例120 631 trap move -3)
速度参考: -3较慢, -9较快 (对应lv0的两种速度)

.....
1128 675
1152 675
1176 675
1200 675

Accepted

1145 645 ac 60 645 920 170



在坐标之后附加 ac 获得一个 ac 方块, 在地图设计的新手阶段不建议添加多个ac。
在ac之后可以继续添加坐标, 作为第二三四...次ac出现的位置参数

游戏地图坐标及地图读入原理

*选学内容：坐标判定 main#42

```
def terrain_check(x,player):
    try:
        global terrainx, terrainy #我们写入的地形
        for i in range(len(terrainx) - 1):
            if terrainx[i] <= x < terrainx[i + 1]: #查找player的x坐标位置，对应的编号
                try: #返回一个此x处对应的地形数组，如果此处没有方块将进入例外处理
                    for j in range(len(terrainy[i]) - 1): #如果此处多于一个可选的y坐标
                        if player.midtop[1] + 5.5 <= terrainy[i][j] < player.midbottom[1]:
                            return terrainy[i][j] #检查player.y位于哪个区间并返回
                    return terrainy[i][-1] #不在所有区间内，返回最下面的位置
                except TypeError:
                    return terrainy[i][0]
    except:
        return 800 #此处没有方块（返回y=800作为下边界），直接掉出边界，角色死亡
```

游戏地图坐标及地图读入原理

*选学内容：坐标判定 main#42 课后习题

请问以下方式构建的墙为什么无法完成阻挡行为？要如何修改？

560 524

584 524 #水平台阶

608 524

608 500

608 476

608 452

608 428 #向上延伸

608 404

608 404 #水平台阶

level函数解析 以level0为例

隐藏地图函数（注意python缩进，这部分相当于直接作为py文件执行）

```
def draw_hidden_map_{等级}():  
    global flag1,flag2,flag3,flag4  
    if flag1==0 and player.x>=300:  
        load_map('./level/level0/hidden_trap.txt')  
        flag1 = 1  
  
    if flag3==0 and flag2==1 and player.x>=980:  
        load_map('./level/level0/hidden_trap_2.txt')  
        flag3 = 1
```

flag参数，作为一次性读入地图的标志，
连续读入会导致游戏卡顿

格式：

```
def draw_hidden_map_等级():  
    global flag  
    if 待读入地图的flag==0 and 前置地图条件 and 玩家位置条件:  
        load_map('./level/level{等级})/待读入地图文件名.txt')  
        待读入地图的flag=1 #防止多次读入
```

level函数解析 以level0为例

初始化函数，在每一次重新开始关卡时执行

```
def initial_{等级}():  
    global g_start,...\  
        flag1, flag2 , flag3 , flag4 , ... , ac_time  
    player.x = player.y = 0  
    player.isalive = 1  
    player.score = 100  
    ...  
    blocks=[]  
    acblocks=[]  
    trapblocks=[] #初始化所有方块  
    terrainx=[(- 32 + 24 * i)for i in range(54)]  
    terrainy=[[] for i in range(53)] #初始化地形  
    #前面的部分建议不要修改  
    load_map('./level/level{等级}/初始地图.txt')  
    ac_time = 0 #初始化碰到ac的次数  
    flag1 = flag2 = flag3 = flag4 = 0 #初始化用到的flag
```

在设计地图的新手阶段

只需将你所用的所有flag初始化为0
并注意导入初始地图的文件
其余部分无需改动

level函数解析 以level0为例

文字提示（嘲讽）函数 它会在游戏时一直执行，在这里写代码相当于在update里写

```
def ac_hints_{等级}():  
    if ac_time == 1:  
        screen.draw.text("...", (675, 300), color='black', fontsize=30, fontname='dpcomic')  
    else:  
        screen.draw.text("...", (300, 100), color='black', fontsize=40, fontname='dpcomic')  
    if flag2 == 1 and flag4 == 0:  
        screen.draw.text("....", (300, 150), color='black', fontsize=40, fontname='dpcomic')  
        screen.draw.text("...", (300, 200), color='black', fontsize=40, fontname='dpcomic')  
    elif flag1 == 1:  
        screen.draw.text("...", (300, 150), color='black', fontsize=40, fontname='dpcomic')  
        screen.draw.text("...", (300, 200), color='black', fontsize=40, fontname='dpcomic')
```

格式

```
def ac_hints_{等级}():
```

```
    if 条件:
```

```
        screen.draw.text("文字（暂只支持英文）", 二元集合坐标, color='颜色', fontsize=字  
体大小, fontname='dpcomic')
```

#dpcomic为我们游戏自带的字体，您也可以向字体文件夹中添加您喜欢的字体

level函数解析 以level0为例

触碰ac次数函数 它会在游戏时一直执行，在这里写代码相当于在update里写

```
def ac_is_hit_{等级}():  
    global acblocks, ac_time, game_state, bullets  
    for b in acblocks:  
        if b.colliderect(player):  
            ac_time += 1 #在此处使得ac_time函数增加1  
            if ac_time <= 1:  
                b.center = [int(b.tag2.pop(0)),int(b.tag2.pop(0))] #上过cb老师课的同学应该看得懂这  
                                                                    #是什么意思doge...好吧，就是把  
                                                                    #ac方块后面的函数pop掉前两个  
            sounds.blip.play()  
        elif ac_time == 2:  
            game_state = 'pass' #过关
```

#注意此处过关所需的ac次数要对应
#这里可以用例外处理简化代码，留作习题 (bushi)

地图设计常见编译错误来源

1. 文件名和函数名必须按照固定的格式编写，尤其是 `levelfunction{等级}` 中四个函数中等级必须修改成对应的等级。
2. `ac` 次数需要与 `txt` 文本中的 `ac` 后跟随数组相匹配。
3. 地图导入时文件请采用文件系统相对路径展示，防止向好友分享您的地图时产生编译错误。
4. 如果游戏出现卡顿，请检查是否未在地图导入时将标志变量改变导致重复导入

建议先微调 `level0` 的文件以熟悉编译操作之后再设计自己的地图