

工作记录

昨天本想直接完成车辆打印的任务，但是最终效果是不如人意，车辆位置漂移严重，需要重新实验观察计算相关位置，同时按照习师兄做法，做 10 车位的停车场，数据集由小到大

今天的任务分为如下几个点：

- 1.进行车辆位置设置：实验计算出合理的位置，生成 10 车位停车场数据集
- 2.进行代码调试，完成任意位置，任意车位组车辆设置
- 3.优化生产速度，完成昨天未完成任务

具体实现：

1.车辆位置设置生成 10 车位停车场数据集

a)由于车辆正反方向位置不同，需要建立 2 套位置，建立后没有成功，后续查看 blender 模型中的位置，只需要一套即可，车辆正反和车型对位置没有任何关系

b)模型非平面，z 平面上，有些目标会陷入地面，有些目标会高于地面

注：为了最终的效果，决定手动调整，先生产出需要的数据集图片

多次调整和数据打印后，仍让不行，打印车辆没有按照打印规则排列

c)尝试直接设置 location 的方式

```
bpy.data.objects[car_name].location=[Car_num_ten_Pos[0][0],Car_num_ten_Pos[0][1],0]
```

没有打印图片，改为()元祖方式后，依旧无法打印

d)添加大量后台 log 打印的代码，分析运行的打印过程后，发现问题所在，如下图所示：

手动建立的车位数组：

```
Car_num_ten_Pos=[
    [-14.2,11.21],[-11.45,11.21],[-8.71,11.21],[-6.2,11.21],[-3.45,11.21],
    [-14.2,5.95],[-11.45,5.95],[-8.71,5.95],[-6.2,5.95],[-3.45,5.95]
]
```

实际打印的结果：

```
parkinglot_line.py x parking_line22.py x chain.py x annotations.txt car_position.txt x
vec_rot: <Vector (-11.4386, -14.0165, 0.0000)> car_name: FIAT500
vec_rot: <Vector (1.4632, -23.0542, 29.7311)> car_name: VWGolfMK4
vec_rot: <Vector (-0.4455, -6.1840, 11.2100)> car_name: VWTouareg.002
vec_rot: <Vector (-4.9011, -14.5937, -0.2143)> car_name: AudiA8.001
vec_rot: <Vector (6.0603, 11.3907, -0.1728)> car_name: AudiA8.004
vec_rot: <Vector (6.5288, 8.2840, -0.1314)> car_name: AudiA8.002
vec_rot: <Vector (-5.9500, -6.2000, 0.0000)> car_name: FIAT500.004
```

注：无法和设置的位置进行匹配，最终发现问题处在设置位置的函数中：

```
def choose_car_position_byname(name, posi_x, posi_y, posi_z=0):
```

具体实现为：

```
def choose_car_position_byname(name, posi_x, posi_y, posi_z=0):  
    bpy.data.objects[name].hide_render = False;  
    vec = mathutils.Vector((posi_x, posi_y, posi_z))  
    inv = bpy.data.objects[name].matrix_world.copy()  
    inv.invert()  
    # vec aligned to local axis  
    vec_rot = vec * inv  
    #annot_file.write(vec_rot)  
    annot_file.write('\tvec_rot:\t'+str(vec_rot)+'\t')  
    bpy.data.objects[name].location = vec_rot  
    bpy.data.objects[name].show_bounds = True  
    return bpy.data.objects[name]
```

本处获取到设置的位置后进行了 inv 的变换，导致最终位置的错误
进行代码更改：

```
def choose_car_position_byname(name, posi_x, posi_y, posi_z=0):  
    bpy.data.objects[name].hide_render = False;  
    vec = mathutils.Vector((posi_x, posi_y, posi_z))  
    inv = bpy.data.objects[name].matrix_world.copy()  
    inv.invert()  
    # vec aligned to local axis  
    vec_rot = vec * inv  
    #annot_file.write(vec_rot)  
    annot_file.write('\tvec_rot:\t'+str(vec_rot)+'\t')  
    bpy.data.objects[name].location = vec  
    bpy.data.objects[name].show_bounds = True  
    return bpy.data.objects[name]
```

最终可正确运行

实现的方法：

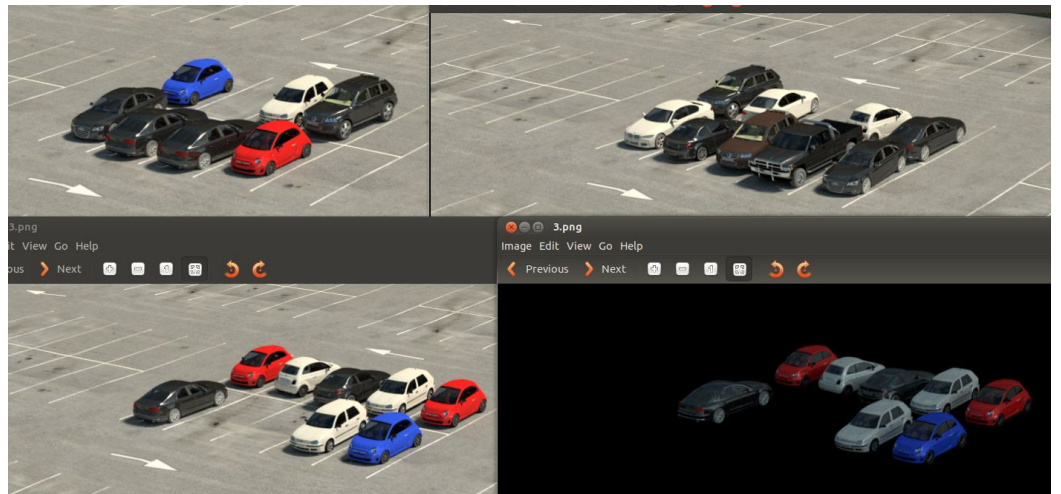
```
#func_2  
Car_probability=0.8  
for car_posi_num in range(0,10):  
    if setCarSpace_random_empty(Car_probability)==1:  
        car_name=car_classes_print_list[car_posi_num]  
        choose_car_position_byname(car_name, Car_num_ten_Pos[car_posi_num][0], Car_num_ten_Pos[car_posi_num][1])  
        #bpy.data.objects[car_name].location=[Car_num_ten_Pos[0][0], Car_num_ten_Pos[0][1], 0]  
        annot_file.write('car_name:\t'+car_name+'\t')  
    else:  
        print('it is not print \n')
```

最终效果图





注：中间多次更改模型，没有将背景换成白色，只需要更改背景图片即可



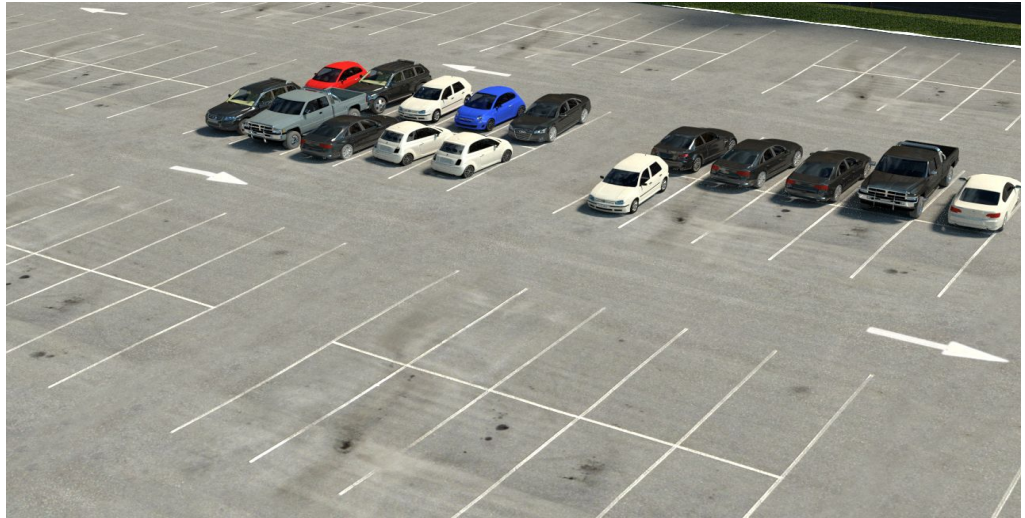
2.代码调试，完成任意位置，任意车位组车辆设置

- i. 昨天已经实现了任意位置的车辆打印，但是出现位置飘逸现象，今天已经被解决
- ii. 继续调试昨天的代码，更改相关函数接口，更加方便调用
 1. 不同实现的方法变成类，直接调用
 2. 公共方法编程类，直接调用

代码贴图：

```
#func 1
Car_probability=0.8
posi_z=0
print(str(car_classes_print_list)+'\n'+str(car_space_posi_list)+'\n')
print(str(car_space_posi_list)+' '+str(Car_print_rows)+' '+str(Car_print_ranks))
for car_space_posi_list_len in range(0,len(car_space_posi_list)):
    for car_posi_print_row_len in range(0,Car_print_rows):
        for car_posi_print_rank_len in range(0,Car_print_ranks):
            print('is is '+str(Car_print_rows*Car_print_ranks+car_posi_print_row_len*Car_print_ranks+car_posi_print_rank_len)+
                  car_name=car_classes_print_list[Car_print_rows*Car_print_ranks*car_space_posi_list_len+car_posi_print_row_len*Car_
                  car_print_posi=getCarSpace_Position(car_space_posi_list[car_space_posi_list_len][0],car_space_posi_list[car_space_
                  annot_file.write(str(car_name)+' '+str(car_print_posi)+' ')
            #print(str(car_space_posi_list_len)+' '+str(car_posi_print_row_len)+' '+str(car_posi_print_rank_len)+'\t'+str(ca
            posi_x=car_print_posi[0]
            posi_y=car_print_posi[1]
            choose_car_position_byname(car_name, posi_x,posi_y,posi_z,Car_probability)
```


打印图片贴图：



3. 优化速度

1. 只 render 打印需要的 OBJECT：已实现
2. 缩小大小的图片：1080P ----> 720P （减少一半的打印时间）
3. 图片剪切组合：暂未完成

明日任务：

1. 继续加速模型
2. 构建深度层，打印深度信息图片（重点）
3. 阅读停车场检测相关文献