

stations

StationReference = Name | MayBelInitial | Platforms

| Name | MayBelInitial | Platforms |
|-----------------|---------------|------------------|
| a = 武汉东站城际场 | 1 | 5, 1, 6, 4, 2, 3 |
| b = 武汉东站存车线 | 1 | 1 |
| c = 余花联络线花山南方向 | 1 | 1, 2 |
| d = 余花联络线余家湾方向 | 1 | 2, 1 |
| e = 武咸城际咸宁南方向 | 1 | 1, 2 |
| f = 武昌南环线武昌南方向 | 1 | 1, 2 |
| g = 武汉东站普速场(没车) | 1 | 7, 4, 2, 1, 3, 5 |
| h = 武昌南环线何刘方向 | 1 | 2, 1 |

timetable

Format: ReportingNumber TrainType MaxSpeedKmph TrainComposition Flags : StationVisit1 StationVisit2

...

StationVisit format:

StationReferencePlatformNumberFromDurationMinutes

TrainType format:

COMMUTER | FREIGHT | IC | URBAN

TrainComposition format:

vvv...

Each v represents one vehicle. L = locomotive (or control post), C = cargo car, P = passenger car

Flags format:

ff

Each f is one flag. 0 = flag not set, 1 = flag set, X = position not used

Flag positions:

1 unused (X)

2 NoBrakingPenalization - if set (1), train does NOT receive penalization when braking at signals

例子

|列车编号|类型|最高时速|动拖布置|flag|车站1|车站2|车站3|

|--|--|--|--|--|--|--|

|C5627| COMMUTER| 200| LPPL| X1 :| b#1#07:03:00#0| a#3#07:06:00#20 |c#1#07:29:00#0 |

车次相关信息

车次信息不变 默认commuter

依照类型设置最高速度 D->200 G->300

车辆细节 先默认 MTM 后续修改长编或者重连?

停站相关信息

仅三点式 进场车站 停站 离场车站

车站 筛选所有车次始发终到，设置字典更改进场离场

股道指定 随机指定?

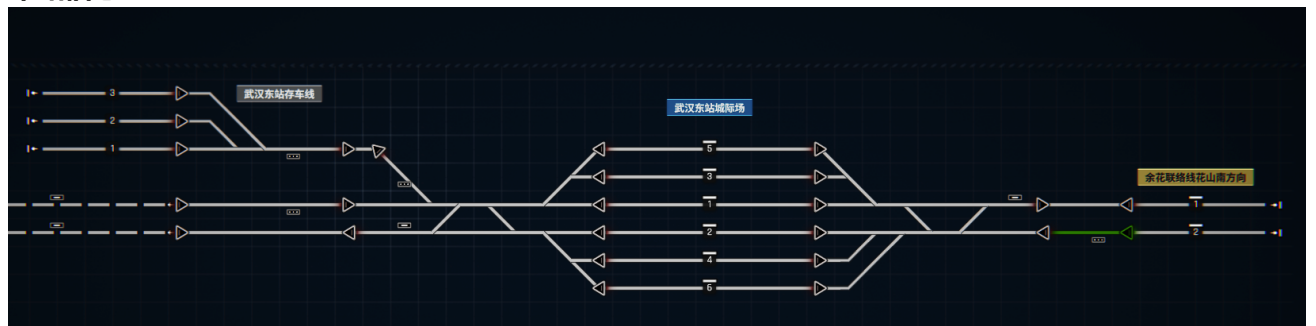
表格信息

从路路通截图导出为Excel进行处理

| | A | B | C | D | E | |
|---|-------|---------|---------|-----|-----|--|
| 1 | 车次名称 | 到时 | 开时 | 始发站 | 终到站 | |
| 2 | C5627 | 7:06:00 | 7:26:00 | 武汉东 | 黄冈东 | |
| 3 | D5770 | 8:10:00 | 8:17:00 | 大冶北 | 云梦东 | |
| 4 | D5762 | 9:15:00 | 9:25:00 | 黄冈东 | 仙桃 | |
| 5 | C5051 | 9:32:00 | 9:36:00 | 咸宁南 | 阳新 | |

始发站和终到站更换为进场车站和离场车站

车站信息



单个车站及进场离场布置

库函数部分及常规列车信息部分

导入库函数，处理表格以及替换字符并导出

使用字典建立常规信息的映射关系

```
In [ ]: import numpy
import pandas
import datetime
import random

Excelpath = "whd.xlsx" # Excel时刻表文件路径
TextPath="trains武汉东.txt" #游戏时刻表文件路径
speed = {'K': '120', 'T': '140', 'Z': '160',
         'D': '200', 'C': '200', 'G': '300'} # 速度映射关系
# 车站-编号,股道,股道到达及用时映射关系
# 股道到达映射关系.可到达股道,图片左右侧线路key值相同则掉向,
station = {'余花联络线余家湾方向': ['d', '135', 0, 10], '武咸城际咸宁南方向': ['e', '135', 0, 10],
          '武汉东站存车线': ['b', '123456', 0, 3], '余花联络线花山南方向': ['c', '246', 1, 3], }
ThisStation = '武汉东站城际场'
# track = {}
# 车型关系--待筛选,默认短编
marshalling = {}

# 始发终到映射关系--待筛选
dst = {}

# 股道到达,图片左右侧线路,,key值不同则掉向
# turnst = {'武九客专武汉方向': 0, '武九客专鄂州方向': 1, '武冈城际黄冈方向': 1}

trainDF = pandas.DataFrame(
    columns=['列车编号', '类型', '最高时速', '动拖布置', 'flag']) # 列车整体信息
arriveStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间']) # 进场信息
stopStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间']) # 停站信息
leaveStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间']) # 离场信息

# 存储字符串形式的最终结果
trainList = []
arriveStList = []
stopStList = []
leaveStList = []
```

读取表格及数据处理部分

- 1,读取表格并去除空行
- 2,从车次提取速度等级
- 3,筛选始发站终到站建立替换字典

车次字符串生成部分

1,车次信息部分

车次号,类型,动拖布置及flag保持默认
速度等级依照KTZDG等区分映射

2,停站部分

停站2(主要车站) 到时及停站时间来自表格部分 股道随机
进场及离场部分 依照游戏先期测试进行平移推算 股道依照上下行安排

```
In [ ]: # 读取文件
sheet = pandas.read_excel(io=Excelpath)

sheet = sheet.dropna()
sheet=sheet.reset_index(drop=True)

print(sheet.head())

trainInfo = sheet["车次名称"].to_frame() # 车辆信息
#统计始发终到车站信息
ts=pandas.concat([sheet["始发站"].value_counts(),sheet["终到站"].value_counts()])
totalStation=ts.index

print(set(list(totalStation))) #所有始发站和终到站统计
```

| | 车次名称 | 到时 | 开时 | 始发站 | 终到站 |
|---|-------|----------|----------|-----|-----|
| 0 | C5627 | 07:06:00 | 07:26:00 | 武汉东 | 黄冈东 |
| 1 | D5770 | 08:10:00 | 08:17:00 | 大冶北 | 云梦东 |
| 2 | D5762 | 09:15:00 | 09:25:00 | 黄冈东 | 仙桃 |
| 3 | C5051 | 09:32:00 | 09:36:00 | 咸宁南 | 阳新 |
| 4 | D5862 | 09:42:00 | 09:46:00 | 阳新 | 咸宁南 |

{'武汉东', '希水南', '汉口', '黄冈西', '黄冈东', '咸宁南', '大冶北', '阳新', '武穴北', '蕲春南', '云梦东', '仙桃', '宜昌东', '十堰东'}

```
In [ ]: # 手动建立映射关系
ArrLeaveSt = {'武汉东': '武汉东站存车线',
              '希水南': '余花联络线花山南方向',
              '汉口': '余花联络线余家湾方向',
              '黄冈西': '余花联络线花山南方向',
              '黄冈东': '余花联络线花山南方向',
              '咸宁南': '武咸城际咸宁南方向',
              '大冶北': '余花联络线花山南方向',
              '阳新': '余花联络线花山南方向',
              '武穴北': '余花联络线花山南方向',
              '蕲春南': '余花联络线花山南方向',
              '云梦东': '余花联络线余家湾方向',
              '仙桃': '余花联络线余家湾方向',
              '宜昌东': '余花联络线余家湾方向',
              '十堰东': '余花联络线余家湾方向'}

ArrLeaveSt
```

```
Out[ ]: {'武汉东': '武汉东站存车线',
         '希水南': '余花联络线花山南方向',
         '汉口': '余花联络线余家湾方向',
         '黄冈西': '余花联络线花山南方向',
         '黄冈东': '余花联络线花山南方向',
         '咸宁南': '武咸城际咸宁南方向',
         '大冶北': '余花联络线花山南方向',
         '阳新': '余花联络线花山南方向',
         '武穴北': '余花联络线花山南方向',
         '蕲春南': '余花联络线花山南方向',
         '云梦东': '余花联络线余家湾方向',
         '仙桃': '余花联络线余家湾方向',
         '宜昌东': '余花联络线余家湾方向',
         '十堰东': '余花联络线余家湾方向'}
```

```
In [ ]: #'列车编号','类型','最高时速','动拖布置','flag' 列车整体信息

for index,row in trainInfo.iterrows():
    trainNum=row["车次名称"]
    maxspeed=speed.get(trainNum[0])#有字头的列车
    if maxspeed==None: #最高速度未找到
        maxspeed='120' #普客

    #trainTC=marshalling.get() #动拖布置
    trainDF.loc[index]=[trainNum,"COMMUTER",maxspeed,"LPPL","X1"]
    trainList.append("{0} {1} {2} {3} {4} : ".format(trainNum,"COMMUTER",maxspeed,"LPPL","X1"))
trainDF.head()
```

Out[]:

| | 列车编号 | 类型 | 最高时速 | 动拖布置 | flag |
|---|-------|----------|------|------|------|
| 0 | C5627 | COMMUTER | 200 | LPPL | X1 |
| 1 | D5770 | COMMUTER | 200 | LPPL | X1 |
| 2 | D5762 | COMMUTER | 200 | LPPL | X1 |
| 3 | C5051 | COMMUTER | 200 | LPPL | X1 |
| 4 | D5862 | COMMUTER | 200 | LPPL | X1 |

```

In [ ]: # arriveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#进场信息
# stopiveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#停站信息
# leaveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#离场信息
# stationInfo = sheet[["到时", "开时", "始发站", "终到站"]] # 时间及停站信息

for index, row in sheet.iterrows():
    arriveSt = ArrLeaveSt.get(row["始发站"])
    leaveSt = ArrLeaveSt.get(row["终到站"])

    # 上下行编号区分车辆进场股道为1,2道

    if int(row["车次名称"].strip()[-1]) % 2 == 0:
        arriveTrack = 2 # 偶数位上行车
    else:
        arriveTrack = 1 # 奇数下行
    # 依照掉向分离场股道 数值相同即为同一侧需掉向, 不同即不掉向顺向行驶
    if (station.get(arriveSt))[2] == (station.get(leaveSt))[2]:
        leaveTrack = 3-arriveTrack # 2->1, 1->2
    else:
        leaveTrack = arriveTrack

    # 停站股道, 依照映射随机选择
    stopTrack = random.choice((station.get(arriveSt)[1]))

    at=row["到时"]
    lt=row["开时"]
    # 统一时间格式
    strTime1 = datetime.datetime.strptime(str(at), "%H:%M:%S")
    strTime2 = datetime.datetime.strptime(str(lt), "%H:%M:%S")

    stopTime = (strTime2-strTime1).seconds/60 # 分钟为单位的停站时间

    arrTime1=strTime1-datetime.timedelta(minutes=(station.get(arriveSt))[3])
    #print((str(arrTime1))[10,-1])
    arrTime2=str(arrTime1.strftime('%Y-%m-%d %H:%M:%S'))[-8:]
    arrTime=datetime.datetime.strptime(arrTime2, "%H:%M:%S")

    leaveTime1=strTime2+datetime.timedelta(minutes=(station.get(leaveSt))[3])
    leaveTime2=str(leaveTime1.strftime('%Y-%m-%d %H:%M:%S'))[-8:]
    leaveTime=datetime.datetime.strptime(leaveTime2, "%H:%M:%S")

    # 进场
    arriveStDF.loc[index] = [arriveSt, arriveTrack, arrTime2, 0]
    arriveStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(arriveSt))[0], arriveTrack, arrTime2, 0))
    # 停站
    stopStDF.loc[index] = [ThisStation, stopTrack, row["到时"], stopTime]
    stopStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(ThisStation))[0], stopTrack, row["到时"], int(stopTime)))

    # 离场
    leaveStDF.loc[index] = [leaveSt, leaveTrack, leaveTime2, 0]
    leaveStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(leaveSt))[0], leaveTrack, leaveTime2, 0))

```

```

In [ ]: print(arriveStDF.head(), '\n', stopStDF.head(), '\n', leaveStDF.head())

```

| | 车站名称 | 股道 | 到达时间 | 停站时间 |
|---|------------|----|----------|------|
| 0 | 武汉站存车线 | 1 | 07:03:00 | 0 |
| 1 | 余花联络线花山南方向 | 2 | 08:07:00 | 0 |
| 2 | 余花联络线花山南方向 | 2 | 09:12:00 | 0 |
| 3 | 武咸城际咸宁南方向 | 1 | 09:22:00 | 0 |
| 4 | 余花联络线花山南方向 | 2 | 09:39:00 | 0 |
| | 车站名称 | 股道 | 到达时间 | 停站时间 |
| 0 | 武汉站城际场 | 6 | 07:06:00 | 20.0 |
| 1 | 武汉站城际场 | 2 | 08:10:00 | 7.0 |
| 2 | 武汉站城际场 | 4 | 09:15:00 | 10.0 |
| 3 | 武汉站城际场 | 3 | 09:32:00 | 4.0 |
| 4 | 武汉站城际场 | 4 | 09:42:00 | 4.0 |
| | 车站名称 | 股道 | 到达时间 | 停站时间 |
| 0 | 余花联络线花山南方向 | 1 | 07:29:00 | 0 |
| 1 | 余花联络线余家湾方向 | 2 | 08:27:00 | 0 |
| 2 | 余花联络线余家湾方向 | 2 | 09:35:00 | 0 |
| 3 | 余花联络线花山南方向 | 1 | 09:39:00 | 0 |
| 4 | 武咸城际咸宁南方向 | 2 | 09:56:00 | 0 |

最终车次结果

合并车辆信息和停站信息

导出部分

导出为train.txt手动附加原文件头部之后替换原时刻表文件

```
In [ ]: timeTable =open(file=TextPath,mode="a")
finalRes=[]
for i in range(0,len(arriveStList)):
    tstr="{0}{1} {2} {3} ".format(trainList[i],arriveStList[i],stopStList[i],leaveStList[i])
    finalRes.append(tstr)
    tstr1=tstr+'\n'
    timeTable.writelines(tstr1)

timeTable.close()
finalRes
```

```
Out[ ]: ['C5627 COMMUTER 200 LPPL X1 : b#1#07:03:00#0 a#6#07:06:00#20 c#1#07:29:00#0 ',
'D5770 COMMUTER 200 LPPL X1 : c#2#08:07:00#0 a#2#08:10:00#7 d#2#08:27:00#0 ',
'D5762 COMMUTER 200 LPPL X1 : c#2#09:12:00#0 a#4#09:15:00#10 d#2#09:35:00#0 ',
'C5051 COMMUTER 200 LPPL X1 : e#1#09:22:00#0 a#3#09:32:00#4 c#1#09:39:00#0 ',
'D5862 COMMUTER 200 LPPL X1 : c#2#09:39:00#0 a#4#09:42:00#4 e#2#09:56:00#0 ',
'D5741 COMMUTER 200 LPPL X1 : d#1#09:42:00#0 a#5#09:52:00#6 c#1#10:01:00#0 ',
'D5778 COMMUTER 200 LPPL X1 : c#2#10:13:00#0 a#4#10:16:00#4 d#2#10:30:00#0 ',
'D5742 COMMUTER 200 LPPL X1 : c#2#11:44:00#0 a#2#11:47:00#4 d#2#12:01:00#0 ',
'D5745 COMMUTER 200 LPPL X1 : d#1#12:08:00#0 a#5#12:18:00#6 c#1#12:27:00#0 ',
'D5852 COMMUTER 200 LPPL X1 : c#2#12:20:00#0 a#4#12:23:00#10 b#2#12:36:00#0 ',
'D5765 COMMUTER 200 LPPL X1 : d#1#12:35:00#0 a#1#12:45:00#9 c#1#12:57:00#0 ',
'D5853 COMMUTER 200 LPPL X1 : b#1#12:47:00#0 a#1#12:50:00#11 c#1#13:04:00#0 ',
'D5881 COMMUTER 200 LPPL X1 : e#1#13:12:00#0 a#1#13:22:00#4 c#1#13:29:00#0 ',
'C5023 COMMUTER 200 LPPL X1 : e#1#13:32:00#0 a#3#13:42:00#8 b#2#13:53:00#0 ',
'C5024 COMMUTER 200 LPPL X1 : b#2#13:57:00#0 a#3#14:00:00#20 e#1#14:30:00#0 ',
'C5056 COMMUTER 200 LPPL X1 : c#2#15:27:00#0 a#2#15:30:00#5 e#2#15:45:00#0 ',
'D5886 COMMUTER 200 LPPL X1 : c#2#15:32:00#0 a#2#15:35:00#4 e#2#15:49:00#0 ',
'D5746 COMMUTER 200 LPPL X1 : c#2#15:46:00#0 a#6#15:49:00#4 d#2#16:03:00#0 ',
'D5854 COMMUTER 200 LPPL X1 : c#2#15:42:00#0 a#4#15:45:00#10 b#2#15:58:00#0 ',
'D5766 COMMUTER 200 LPPL X1 : c#2#16:15:00#0 a#4#16:18:00#5 d#2#16:33:00#0 ',
'D5855 COMMUTER 200 LPPL X1 : b#1#16:08:00#0 a#2#16:11:00#10 c#1#16:24:00#0 ',
'D5758 COMMUTER 200 LPPL X1 : c#2#17:04:00#0 a#6#17:07:00#5 d#2#17:22:00#0 ',
'D5856 COMMUTER 200 LPPL X1 : c#2#17:37:00#0 a#4#17:40:00#10 b#2#17:53:00#0 ',
'C5067 COMMUTER 200 LPPL X1 : e#1#17:42:00#0 a#1#17:52:00#4 c#1#17:59:00#0 ',
'D5895 COMMUTER 200 LPPL X1 : e#1#17:50:00#0 a#1#18:00:00#4 c#1#18:07:00#0 ',
'D5857 COMMUTER 200 LPPL X1 : b#1#17:57:00#0 a#4#18:00:00#11 c#1#18:14:00#0 ',
'D5777 COMMUTER 200 LPPL X1 : d#1#18:34:00#0 a#5#18:44:00#13 c#1#19:00:00#0 ',
'D5769 COMMUTER 200 LPPL X1 : d#1#19:23:00#0 a#1#19:33:00#8 c#1#19:44:00#0 ',
'D5761 COMMUTER 200 LPPL X1 : d#1#19:33:00#0 a#1#19:43:00#5 c#1#19:51:00#0 ',
'C5068 COMMUTER 200 LPPL X1 : c#2#19:37:00#0 a#2#19:40:00#6 e#2#19:56:00#0 ',
'D5896 COMMUTER 200 LPPL X1 : c#2#19:43:00#0 a#2#19:46:00#4 e#2#20:00:00#0 ',
'D5858 COMMUTER 200 LPPL X1 : c#2#20:19:00#0 a#4#20:22:00#10 b#2#20:35:00#0 ',
'D5859 COMMUTER 200 LPPL X1 : b#1#20:37:00#0 a#1#20:40:00#11 c#1#20:54:00#0 ',
'D5757 COMMUTER 200 LPPL X1 : d#1#20:54:00#0 a#1#21:04:00#4 c#1#21:11:00#0 ',
'C5512 COMMUTER 200 LPPL X1 : c#2#21:36:00#0 a#4#21:39:00#4 d#2#21:53:00#0 ']
```

In []: