

新增函数

1. `Route, Lenth = get_path((x1,y1),(x2,y2))` 获取最短路径以及长度
2. `return_metrics()` 返回指标

Observation

```
obs = [ {无乘客车辆}、{有一个乘客的车辆}、{有两个乘客的车辆}、{匹配不成功乘客}、{匹配成功乘客} ]
```

```
{无乘客车辆}
```

```
- key : driver_id  
- value: ([], (x,y))
```

```
{有一个乘客的车辆}
```

```
- key : driver_id  
- value: ([乘客列表], (x,y))
```

```
{有两个乘客的车辆} # 这个先传给你，你可以先放着不用，我怕后面计算可能会用得到
```

```
- key : driver_id  
- value: ([乘客列表], (x,y))
```

```
{匹配不成功乘客}
```

```
- key : passenger_id  
- value: ([], (x,y))
```

```
{匹配成功乘客} # 这个先传给你，你可以先放着不用，我怕后面计算可能会用得到
```

```
- key : passenger_id  
- value: ([对应车辆ID], (x,y))
```

Action

```
action = [ {匹配成功的字典}, [未匹配成功的车辆列表], [未匹配成功的乘客列表] ]
```

```
{匹配成功的字典}
```

```
- key: driver_id  
- value: passenger_id
```

```
[未匹配成功的车辆列表] # 这块为了方便，如果可以的话麻烦也传给我一下，如果太麻烦就算了
```

```
= [driver_id, driver_id, driver_id, ... ]
```

```
[未匹配成功的乘客列表] # 这块为了方便，如果可以的话麻烦也传给我一下，如果太麻烦就算了
```

```
= [passenger_id, passenger_id, passenger_id, ... ]
```

环境设定

- 车辆：随机分布 订单数据量的 1/4
- 起始速度：30km/h
- 每 10s 交互一次