最近需要做的出来的，两个agent(Bus Agent + Policy Agent)，初步实现的就是路网导入，积水点随着时间推进的变化，然后Bus agent在初始路线上行驶。

修改内容：

**1. 地图和空间环境**

现有代码中的网格模型（如 SingleGrid 或 MultiGrid）是基于离散的坐标系实现的，现需使用实际的路网和经纬度数据。需要做如下调整：

* **使用地理坐标系统**：将地图坐标（经纬度）集成到环境模型中。可以创建一个新的类或扩展现有的网格模型，支持处理经纬度数据，或直接使用基于地理信息系统（GIS）的数据模型。

例如，可以使用 GeoDataFrame 或类似的工具来处理实际的道路网络，或者通过自定义 Position 类型（经纬度）来实现智能体在地图上的定位。

* **地图数据导入**：可以导入实际的路网数据，使用如 OpenStreetMap 或高德地图 API 获取路网数据。需要将路网转换为适合在模型中使用的格式，如图（networkx）或自定义网格。
* **更新网格类**：如果选择基于图（例如 NetworkGrid），需要适配路网节点（例如，路口）和边（道路）为 Agent 的可能位置，并支持基于经纬度进行位置计算。

**2. 智能体模型**

两种类型的智能体：**Bus Agent** 和 **Policy Agebt**。可以基于现有的 Agent 类来创建这些智能体，然后给他们设计特定的行为逻辑。

* **公交车智能体（Bus Agent）**：
  + 在 Agent 类的基础上，创建一个 BusAgent 类，并为其添加如 current\_position（当前位置）、route（行驶路线）等属性。
  + step 方法可以模拟公交车的动态行为，例如，公交车根据路线行驶、停靠等。
  + move 方法可以根据调度策略和路网信息来控制公交车的移动。
* **政策智能体（Policy Agent）**：目前只让他存在就可以
  + Policy Agent 类用于制定公交车的调度策略，基于内涝积水点的信息和实时路况来调度公交车的行动。
  + step 方法可以定义如何根据当前积水点、天气情况、交通状况等因素制定决策。

**3. 内涝积水点数据**

积水点数据是基于实际经纬度的，需要处理这些数据：

* **积水点数据管理**：可以将积水点数据存储在一个类似 Grid 或 NetworkGrid 的数据结构中，结合其经纬度信息。需要根据经纬度和路网数据更新积水点的状态，并影响到智能体的决策。
* **数据收集和决策**：将积水点信息（深度、影响范围等）作为输入，供政策智能体进行决策。可以通过调整现有的 DataCollector 类来收集每个时间步的积水点信息，并将其与公交车的调度行为进行关联。