本次迭代可以直接基于第2次迭代的代码进行，与第3、4次迭代在仿真引擎上的改进增强实际无关。因为在第2次迭代中很好的分隔了显示部分、存储部分和仿真引擎部分，因此可以用其它技术来实现显示。

本次迭代的目标是将原仿真引擎部分改造为基于SpringBoot的仿真引擎服务，并提供管理用RESTful接口（启动、停止）和获取仿真中各对象状态数据的RESTful接口（即第2次迭代中读取DisplayQueue的接口）。

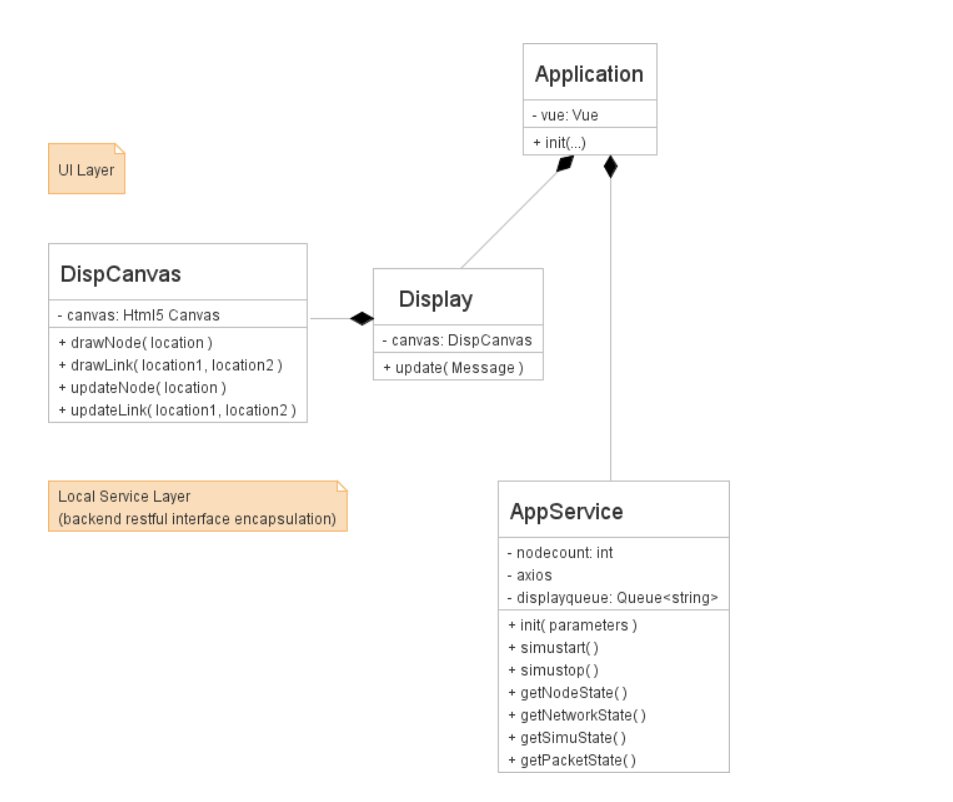
要求

R1：而数据显示部分则直接写成单页面应用，用javascript、html5 canvas（直接代替Display中的canvas）开发。推荐引入vue.js以方便人机交互功能的开发。对这部分没有认识的同学，可以参考vue网站上的教程进行学习。但是因为我们的应用十分简单，所有无需使用web pack管理前端工程也没问题。

R2：页面上分配一个矩形区域（canvas）并提供绘制数据功能。（代替源代码中Display类的canvas或graphics）

R3：提供两个按钮：Start和Stop。

R4：提供一个文本框，允许输入一个数，作为仿真中结点的个数。



注1：R3、R4可引入vue.js会非常方便，调用服务端RESTful接口可以用axios或其它成熟稳定的vue领域的组件。

至于R2，实际跟vue无关。但是也可以用vue管理。参考 [vue 中使用canvas 画图 - 知乎 (zhihu.com) (Links to an external site.)](https://zhuanlan.zhihu.com/p/385020615#:~:text=vue%20%E4%B8%AD%E4%BD%BF%E7%94%A8canvas%20%E7%94%BB%E5%9B%BE%201%201%E3%80%81html%202%202%E3%80%81js%20%28base64%E8%BD%AC%E6%8D%A2%E4%B8%BA%E6%96%87%E4%BB%B6%29,3%E3%80%81%20%E5%85%B7%E4%BD%93%E7%94%BB%E5%9B%BEjs%204%204%E3%80%81%E4%BD%BF%E7%94%A8%20vue%205%205%E3%80%81%E4%BD%BF%E7%94%A8%20js)

注2：由于服务端SpringBoot微服务框架的特点，特别是SpringBoot会有自己的线程，所以在合并基于SpringBoot的应用和仿真引擎代码时，强烈建议创建一个新的thread，并在该thread中运行仿真引擎，与SpringBoot应用自己运行所使用的线程分开。但这样的话，要注意消息队列应改为Concurrent Queue，因为此时是存在于两个不同thread的代码在并发的访问该队列。

注3：与桌面版仿真不同的是，此时html应用只能以查询方式从服务端周期性拉数据并用于更新显示。如果需要实时传输，则需要引入WebSockets功能（SpringBoot支持但需要开发者调用）、SockJS（html侧）等。