表现层接收用户输入，点击、拖拽等

拖拽过程中元素跟随手指，在表现层计算

逻辑层应该知道什么

点击

拖拽开始

拖拽结束时的目标位置

拖拽结束时才通知逻辑层

拖拽过程中的视觉反馈处理

对于拖拽时的视觉表现(如 缩放、半透明等)，也应该完全由表现层控制

为什么不应该在逻辑层计算拖拽位置？？

1、破坏了分层架构、逻辑层依赖了 Unity 的 Camera 和输入系统

2.增加了不必要的通信开销(每帧事件发布)

3.使逻辑层难以进行单元测试(因为依赖 Unity 环境)

4.降低了相应速度(需要等待事件循环)

[输入阶段]

手指移动 → View直接更新Transform → 视觉上跟随手指

[逻辑确认阶段]

手指抬起 →

View将最终世界坐标转换为逻辑坐标 →

通知ViewModel →

ViewModel决定最终位置/合并逻辑 →

通过事件通知View最终需要表现的位置/效果





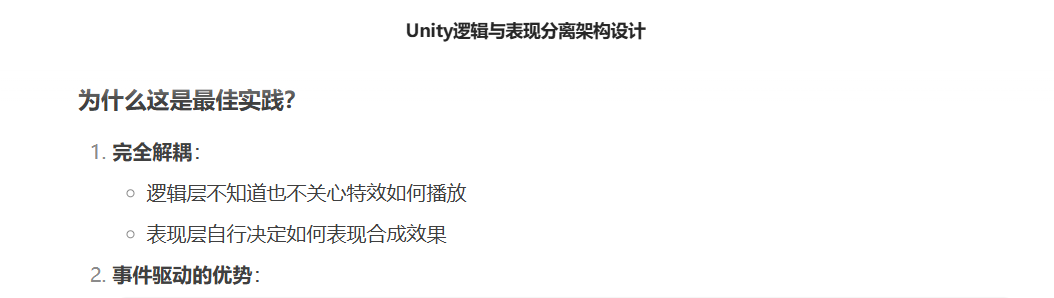


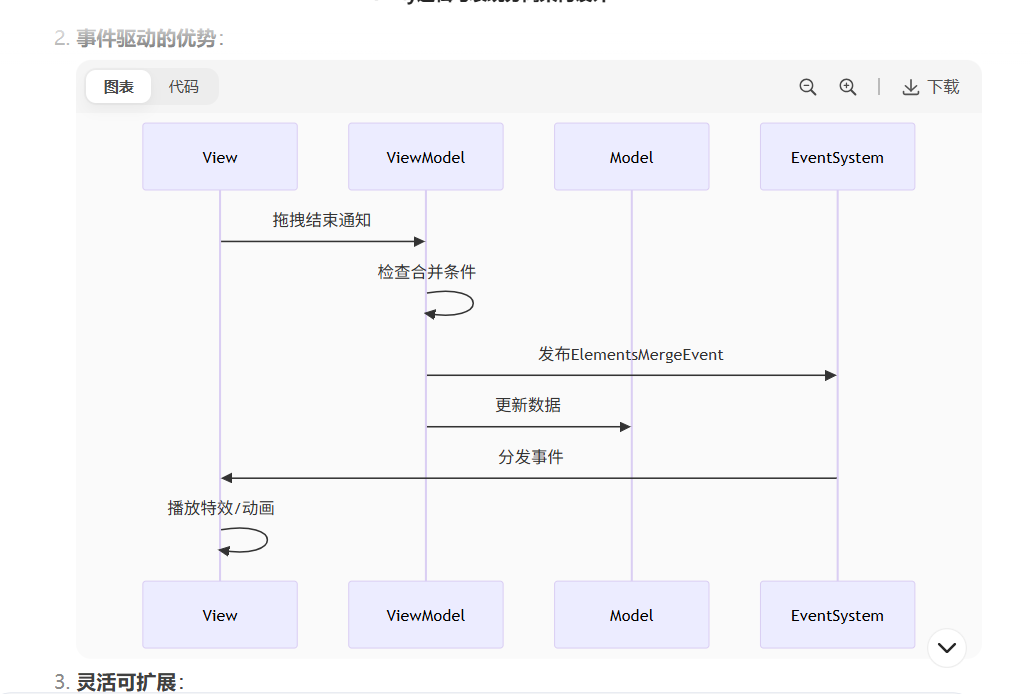


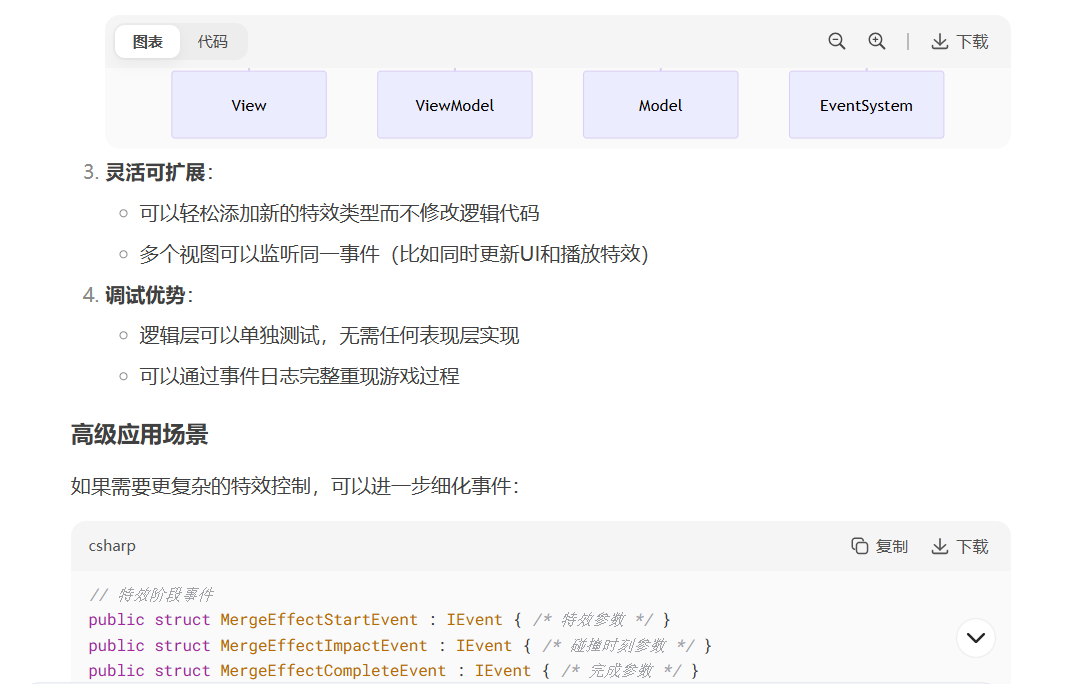




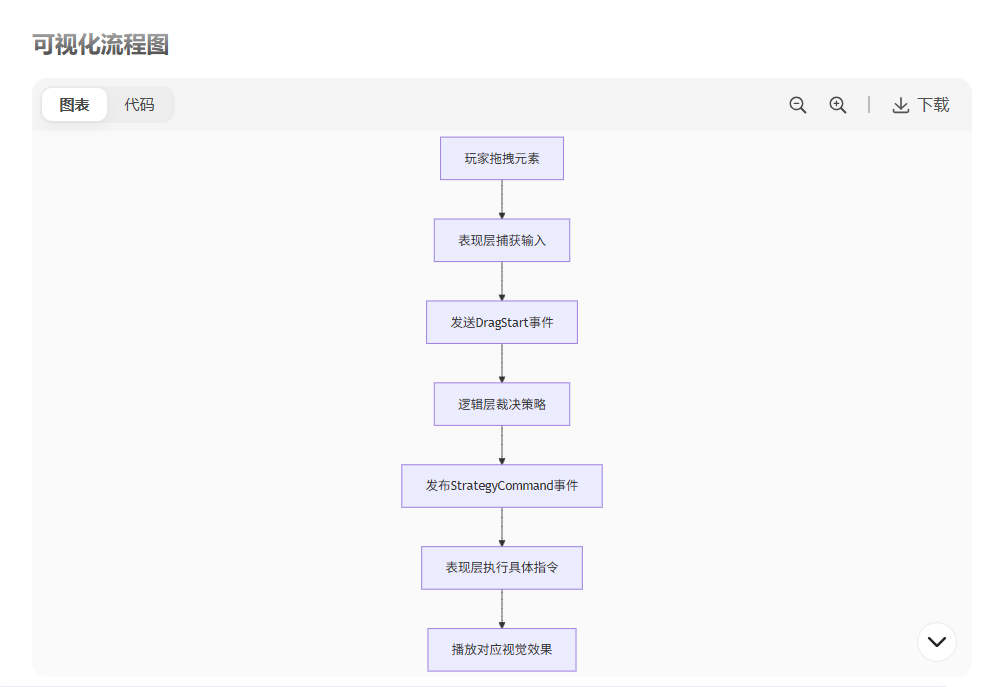


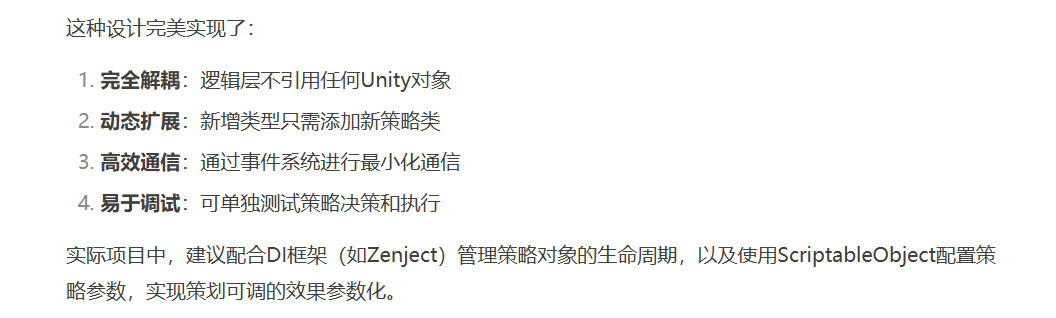












比较好的思路

https://jc-performance.cn//online/1837\_139473050.html

https://www.sohu.com/a/453864546\_667928

https://blog.csdn.net/NRatel/article/details/88825045

已读

https://zhuanlan.zhihu.com/p/28401100

更多的是性能优化相关

https://blog.51cto.com/u\_16213588/11514840