框架的优势（相比《传世手游》）：

1.在层级关系上：

从宏观分成3层（逻辑，桥梁，底层），可以使业务和底层模块尽量低耦合，

就像一辆汽车里（有驾驶员，方向盘，汽车引擎）。

好处：1.人员分工：人员的任务分工明确，彼此关注不同复杂度的模块，互不干扰

2.代码模块化：有利于代码的模块化，当项目中后期复杂度提高时，不易造成混乱

3.代码标准化：标准化和规范化代码产出，容易维护和调试，也容易查阅理解

4.代码重用：不会产生冗余代码，模块切分到适合的程度，可以重用给其他项目

5.逻辑清晰：3层架构中的各个模块，都采用清晰易懂的设计降低原本复杂的设计

比较：目前传世代码尚没有类似的大的分层思路和机制，代码量巨大难以阅读和维护

2.机制的运用上：

使用了Entity机制，组件机制，MVC机制等机制

好处：1.简化了程序员的思考，运用良好机制可以使工作更轻松和高产出，最重要是减少加班。

2.易于调试和优化，项目到中后期业务量上升时可以极大控制错误的出现概率。

3.代码业务清晰：不易有冗余的代码，提高效率

比较：目前传世代码运用的机制很少，传统的面向对象编程思路且设计有限，比如逻辑控制，数据，

显示这样的MVC机制运用程度较低。程序员的产出效率不好说

3.业务替代上：

好处：如果员工离职等原因造成工作内容滞后，良好的架构设计可以让其他人快速的接手

比较：目前传世代码如果发生人员离开，相应的业务接手需要较长时间。

4.上线前压力应对：

好处：项目在中后期代码大规模增长时不易混乱和难以管理，以及各种针对代码的优化都可以聚

焦和定位，各种代码Bug都易于定位。

比较：目前传世代码在应对调优，测试都面临不小的难度。