整体设计思路

首先我觉得没必要这么干：

国产cpu采用的架构和其所对应的指令集

华为鲲鹏、飞腾使用的ARM架构。

海光CPU、兆芯使用的是X86架构。

龙芯采用的是MIPS架构，现在龙芯有自己的指令集。

而申威使用的是alpha架构。

指令集可分为复杂指令集(CISC)和精简指令集(RISC)两部分

ARM和MIPS采用的是精简指令集(RISC)

X86采用的是复杂指令集(CISC)

拿国产飞机举例，不能因为国产发动机推力不够，尚不满足第四代重型战斗机的需求，就不去开展歼-20的设计开发工作吧。现在涡扇十五随着不断地改进升级，也足够承载歼-20了，可谓如虎添翼。而且造的核心是飞机，就算所谓纯国产的飞机，就能保证机床，加工设备，材料加工等要素，都是全国产吗？

模拟仿真也是同样的道理，开发技术和运行环境等客观因素固然重要，但是更重要的是设计模式，理念。设计模式掌握了，随着技术的发展与时俱进。仿真模拟程序的才是核心，无需太关注采用什么技术将其开发实现，也不用太关注要在什么环境平台下运行，这些都是次要的。要关注的是设计思路，仿真效果，真实程度等方面。

如果用unreal engine开发的话，原码大部分开放。做出来的东西完全独立自主可控。

开发环境和运行环境中，cpu得用国外的，操作系统得用国外的。但是随着技术的发展自研cpu，操作系统也得往好的方向发展，逐渐就相匹配了。

如果非要这么干：

介绍

和80前的人讲述这个设计思路，就不能用电子游戏来作类比。他们大部分人是听不懂，也不会接受，不会试图去理解的。他们只会觉得不可理喻。所以针对这类人给他们讲解的时候，就需要把古人智慧的结晶，军事仿真的鼻祖，象棋，围棋拿来说事了。

整体思路

我将仿真模拟分为以下几部分，其中包括表现层，逻辑层，实体层，实体功能层，决策层和算法层。

如果利用游戏引擎进行开发的话，以上所有的部分都可以通过游戏引擎直接编写，实现。而且效果不会很差。如果是要脱离游戏引擎，或者说只将游戏引擎作为表现层的观览工具的话，那实现起来将会非常困难，会面临诸多技术难题。

不论利用哪种方式，我们都需要先对程序的开发进行规划和设计。

表现层（对外展示）

用于展示仿真模拟的过程，展示的方式可以有多种，主要为U3D,UE4,控制台。其中利用游戏引擎作为表现形式的话，优势是可视化，利用渲染引擎，直观的将推演过程展示给我们。缺点是，开发成本高。与其这样还不如直接利用引擎进行开发。

利用控制台展示的话，其实就是利用文字实时输出仿真模拟战况，优势是开发便捷，资源占用小，缺点就是展示不够直观。类似于失传已久的文字冒险类游戏。

逻辑层（主体核心）

程序的主体逻辑部分，包括刷新间隔，场景加载，实体加载，时间线，规则，是整个仿真模拟系统的核心所在。正是因为有逻辑层的存在，才意味着仿真模拟系统的存在。是道的存在。

实体层（主体核心）

创建装备实体，武器实体，弹药实体等，将现实中的军事装备抽象化，根据主题逻辑设计相应的属性。

实体功能层（主体核心）

为实体进行功能开发，包括机动，侦察，瞄准，锁定，打击，维修等功能。利用实体属性，实现仿真的行为表现。

决策层（锦上添花）

相对独立的一部分，也是开发难度最大的一部分，可以先设计一个基础的决策，先让程序再无人操作的状态下运行起来。是一个需要不断优化的部分。

算法层（基础支撑）

提供以上几层所需要用到的运算基础。

对接方式

表现层与逻辑层进行对接，采用进程间通行或网络通信的方式进行。表现层创建的装备对象只需要具备状态信息即可，采用状态同步的机制。

算法层的调用都采用调用动态库或静态库的方式进行。

决策层的调用需要根据实际的开发情况来选择调用方式。一般也是动态库或静态库的方式进行。

逻辑层，实体层，实体功能层做为一个整体部分，直接利用对象，函数调用即可。

使用语言

算法层可以用C语言进行开发，提高效率

决策层还没调查了解，不知道用python好还是直接用C++合适

逻辑层，实体层，实体功能层则可使用C++进行开发

用象棋做比喻

棋子和棋盘的颜色，材质，大小都可以视为表现层。有高级的，有简易的。但他丝毫不会影响象棋本身的进行。表现层只考虑棋子的位置，存在与否即可。

逻辑层就是象棋的规则，是象棋的本体逻辑。

实体层就是每一个独立的抽象的棋子，他们有着自身的属性和方法，比如棋子都具备机动和打击的功能。

实体功能层则是对实体中功能的实现。具体的马走日，车走直线，炮得隔着另一个棋子才能打击等。

决策层相当于人机对战中的电脑。

算法层则是为了以上全部过程提供计算支撑，比如计算是否已经绝杀对方。

工作可以分为以下几个部分：

1. 表现层独立开发，为了满足展示的需求，一组人需要先利用unity进行开发，以完成整套逻辑为目标，注意松耦合，接口设计，预留等问题。
2. 整理装备性能参数数据，并且存入数据库，尽量能搜集多少就搜集多少，最好搜集一些实战演习的实际操作资料，战场资料等。
3. 明确如何抽象战场环境，地图怎么表现，装备怎么表现。

首先先做个二维的，做一个类似于侠盗猎车手1-2那样的。

1. 算法的开发，主要偏功能，比如避障功能，寻路。这些东西是基础，以后版本再怎么升级迭代，这些本质都是不变的。