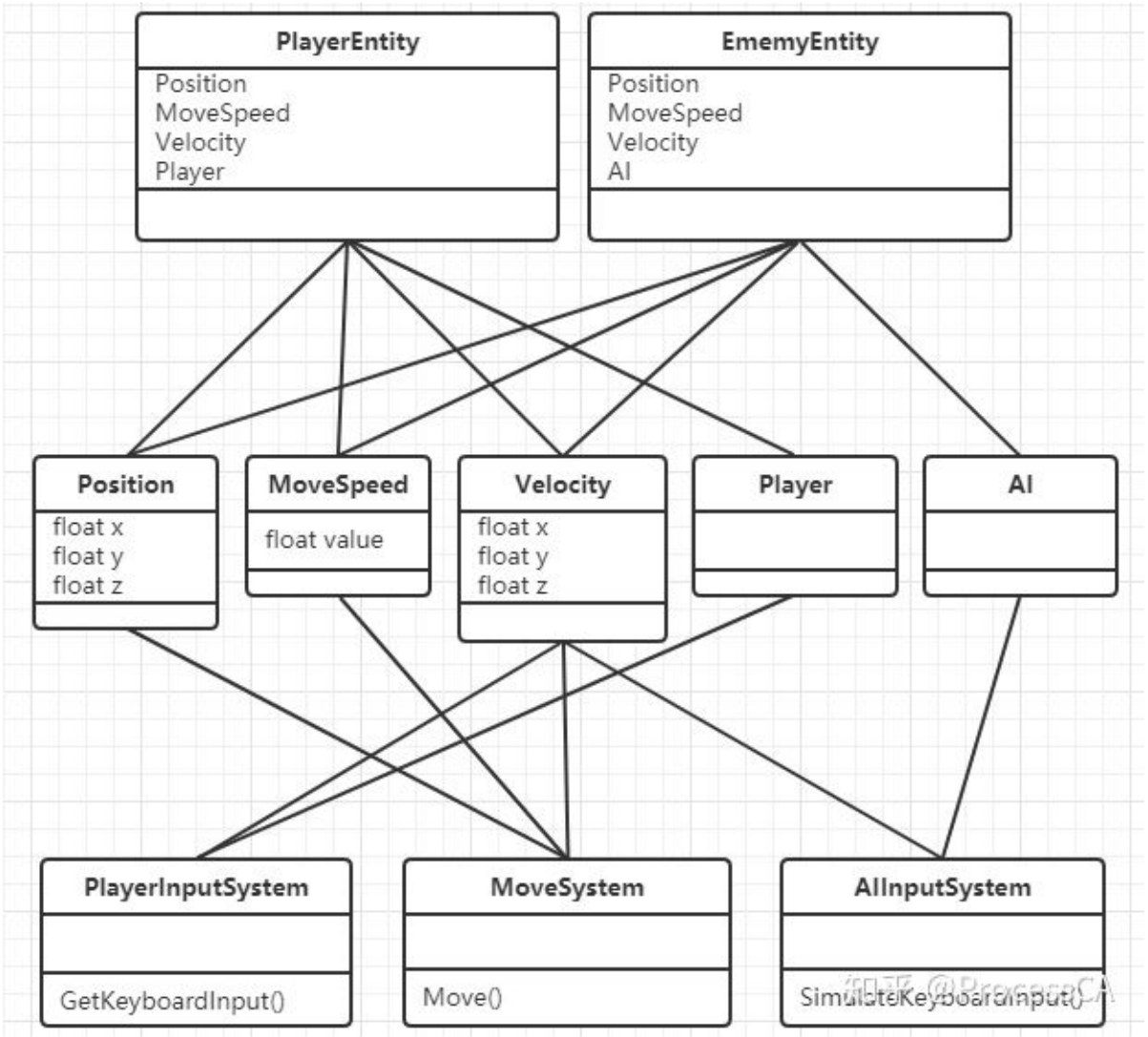
什么是ECS？

- E -- Entity 实体，本质上是存放组件的容器
- C -- Component 组件，游戏所需的所有数据结构
- S -- System 系统，根据组件数据处理逻辑状态的管理器

Componet组件只能存放数据，不能实现任何处理状态相关的函数，而System系统不可以自己去记录维护任何状态。说的通俗点，就是**组件放数据，系统来处理**。这么做的好处，就是为了尽可能地让数据与逻辑进行**解耦**。与此同时，一个良好的数据结构设计，也会以增加**CPU缓存命中**的形式来提升性能表现。



Entities 实体是游戏中的事物

Components 组件时游戏中的数据

System 系统是游戏中的逻辑

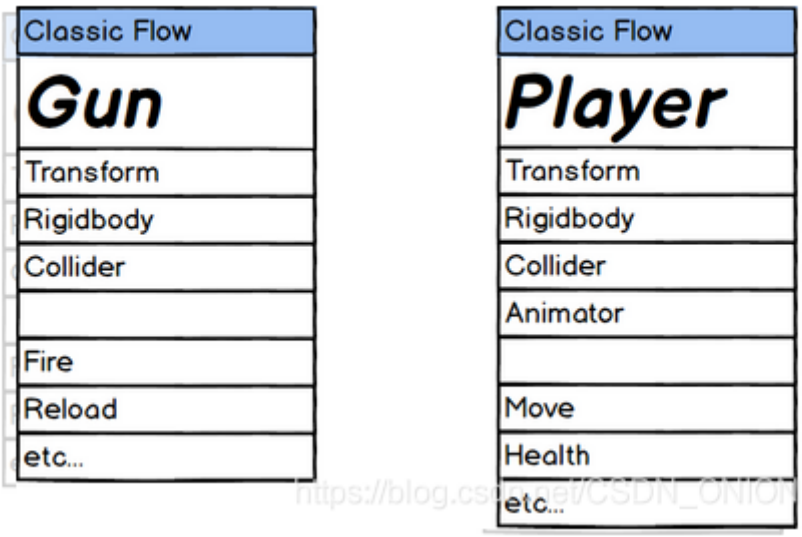
由于ECS架构的一些特点，他可以很容易利用多个CPU实现逻辑并行，紧凑且连续的内存布局，比起OOP可以更方便地获得更大的性能提升。

只需要考虑每一种GameObject所包含的数据实体，而不用考虑自己的组件集合（抛弃了Transfrom，Rigidbody等），将处理与各个对象类型完全分离。实体仅仅是一个句柄（或者说是一个标识符）永远索引它表示的不同数据类型的集合(ComponentDataGroups)系统可以通过这些句柄来对所有组件进行过滤和操作，而不需要将系统与实体类型明确结合。这种工作机制有很大优势，它不仅能提高缓存效率，缩短访问时间，

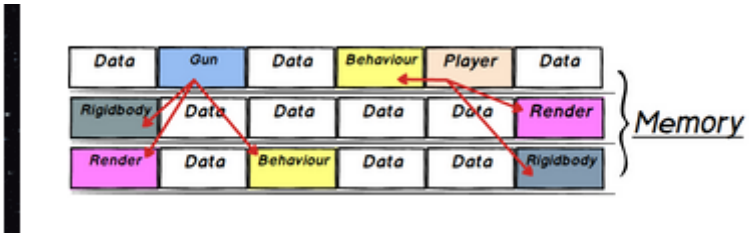
OOP，Object Oriented Programming

DOTS

举个最简单点的例子来区分 面向过程和面向对象
有一天你想吃鱼香肉丝了，怎么办呢？你有两个选择
1、自己买材料，肉，鱼香肉丝调料，蒜苔，胡萝卜等等然后切菜切肉，开炒，盛到盘子里。
2、去饭店，张开嘴：老板！来一份鱼香肉丝！
看出来区别了吗？这就是1是面向过程，2是面向对象。



Gun与Player所引用的Transform、Rigidbody、Collider等这些关键脚本被分散在堆内存中，数据将不会转换成可由更快的SIMD（单指令多数据流）矢量单元进行操作的形态。



这种数据存储方法的随机偶发性质。每一个单引用，在使用时都有可能会将其所有的成员变量从系统内存中全部拉出

面向对象有什么优势呢？首先你不需要知道鱼香肉丝是怎么做的，降低了耦合性。如果你突然不想吃鱼香肉丝了，想吃洛阳白菜，对于1你可能不太容易了，还需要重新买菜，买调料什么的。对于2，太容易了，大喊：老板！那个鱼香肉丝换成洛阳白菜吧，提高了可维护性。总的来说就是降低耦合，提高维护性！

面向过程是具体化的，流程化的，解决一个问题，你需要一步一步的分析，一步一步的实现。

面向对象是模型化的，你只需抽象出一个类，这是一个封闭的盒子，在这里你拥有数据也拥有解决问题的方法。需要什么功能直接使用就可以

了，不必去一步一步的实现，至于这个功能是如何实现的，管我们什么事？我们会用就可以了。
面向对象的底层其实还是面向过程，把面向过程抽象成类，然后封装，方便我们我们使用的就是面向对象了。

举个例子，当我命令Gun进行开火，子弹飞出，对子弹的坐标进行运算让它飞行，表面上看起来仅仅是对子弹这个对象的Transform中的position进行了操作，但实际上，子弹的rotation，gameobject属性，还有等等等等其他成员也一并拉出来操作了。

问：Entities的ForEach是干什么用的？
答：用来筛选实体的。

问：通过什么筛选？
答：通过组件筛选——把包含指定组件的实体筛选出来。