几个主要的静态的东西的关系。

利用了自带的关卡蓝图和gamemode。

然后我自己写了一个静态的，是battle zone framework

然后其他的都是动态spawn出来的。

我在关卡蓝图里面把battle zone framework的引用给到到gamemode里，并且生成并启用了UI，这个ui把场景的渲染结果放到一个1000x1000的图片上，然后其他区域是将来要加分数一类的信息的地方。

然后所有的这些物体可以通过get game mode, cast to 我自己那个gamemode,然后get战斗区主框架，然后get每一个东西。

这些东西包括，关卡管理器，子弹管理器，掉落物管理器（还没写）。

自机的引用暂时是存在这个主框架里面的，估计以后要改，可能会有一个专门的自机管理器。

UI的引用也应该给到主框架。

**////我的命名规范////**

我应该基本是遵守了的，就是所有东西，我会写成什么什么base，然后你们从这个上面派生（derive）就行了。

然后所有的材质是从一个叫 最常用半透明材质 来的。

所有的精灵是一个叫 没有碰撞的quad，用static mesh component，然后x方向转180度。

反正看到没对的直接跟我说，有可能是最早的代码忘了规范了。

**////关卡////**

然后说一下关卡的组织形式。

这一部分主要是要让大家在合作的时候有一个可以通用的框架，这样你可以把你的关卡直接复制到别人的工程里面，然后只需要很少的动作你们就可以把不同的关卡放到同一个游戏，也可以把不同的敌人放到同一个关卡里。

这儿的逻辑是这样的。

关卡管理器

里面存了一个关卡的数组（stage base），在正常运行或者运行调试的时候，管理器负责调用对应的逻辑。

然后关卡里面是存的关卡事件的数组（stage event base），每个关卡事件都有2个变量，一个是等待组，另外一个是触发时间。这儿的逻辑是这样的。如果一个等待组是开启状态，那么在关卡里面会给这个等待组记录总共执行的时间，并且如果这个时间大于了某一个时间的触发时间，就触发，触发的时候是去调用那个事件的一个函数，好像叫触发我。然后你可以任意的控制等待组的开关状态，关闭的时候，计时不会增加，也肯定不会触发时间。

然后关卡里面有一个函数，刷新，会自动的吧关卡事件整理一下，放入不同的关卡事件数组，并且是拍了序的，这样会很方便后面的逻辑。

意思就是说，你们给关卡添加事件的时候，应该是添加到那个总的事件列表里，添加完成了，调用刷新，然后只需要控制等待组的开关，其他统统不要动，除非你很了解这些蓝图的工作原理。

设置等待组的理由。这个概念确实很奇怪，所以我简单说一下。设计初衷是，如果你需要一个事件发生以后，再按正常顺发触发其他的事情，那么你可能是在前面写，调用（call）（可以简单的理解为，调用就是spawn小怪） ，然后delay，然后再call，然后中间停下来，然后写一个custom event，然后再继续call delay的循环。这样有一个问题，你没有办法从中间的一个地方开始调试，而且如果你们要改东西，就要改代码，没有办法静态的加上去。所以我就用了一个不那么优雅的写法，但是至少可以从任何的时间开始调试。那么等待组，你可以理解为，先只开0号组，然后0号组最后会生成5个特殊的小怪，这5个死的时候，这个关卡会计数，计数到5，就是说这5个小兵都死完了，那么关闭0号组同时开启1号组，这样 。这个可以做道中boss之后，需要击破了道中boss然后才会spawn出来的小怪。

暂时这么写吧，后面想到更好的写法了再改。

所以如果你们有写非关底的boss，就要记得控制等待组，这个我是不会帮你们写的。

然后等待组的总数，我默认写的8个（那个变量是最后的一个引用，所以是7），如果需要更多的等待组，自己进去改。

然后Boss里面会存spell card，spell card里面我还会再分一层，叫做spell card event。这个意思是，如果你在某一个时候放了b，然后当前的一个event的子弹会被消弹，之后这个event会被标记为discard，然后这个event就不会再继续放子弹了，这样可以保证所有放出来的子弹的形状都是很漂亮的。也可以让这个event不能被discard，这样就可以一直放子弹，比如你要铺密度的时候，就可以这么干。

使用技巧。

那个spellcard event的写法，你新建蓝图，然后保证继承关系是正确的，然后到基类里面，把他的eventgraph里面的东西复制到你的新的那个里面，然后写。只有这样。

如果你要单独测试一个spellcard event，你可以单独把这个物体放到场景里面，然后你可以在任何地方调用这个spellcard event的start。

**////子弹类////**

子弹类是这样的。基类只给了几个，我觉得应该估计你们自己去创造。

给了一个总的基类，只要你是子弹，就是这个基类的子类，这一层要做的事情是，要管理自己在子弹管理器里面要存一份。后面可能要改这个功能的具体实现，因为好像只和消弹有关，而消弹应该是有别的做法的。

然后这个基类还要对子弹的队伍行为进行规范，就是说所有子弹会带一个表示他的所属队伍的component，这个必须要有。这个component好像是自动填充成敌人的子弹。忘了。

然后从这个基类派生出4个东西，简单弹（normal bullet base），激光（laser base），曲线激光（blent laser base），和玩家的子弹（player bullet base）。

然后是其他的一些东西，你们可以看到还有几个文件夹。

接口，这个暂时没什么用，我是准备在简单弹上面加这个接口，主要控制一下设置初速度和初始角度，具体的很简单，但是可能需要你们自己去手写。

队伍信息，没什么好说的，就是存一下子弹是不是你自己的，这个在4个基础派生类里面已经给你填充好了，直接用就行。

曲线激光的一些组件。曲线激光是这样的，你调用他的一个叫update什么什么的函数，把最新的头部位置给进去就行了，其他的不用管，如果你一定要管，你会发现里面有一个控制节点的数组，应该是一个vector2d的数组，你进去改就可以了。

离屏行为。就是所谓的出屏消弹。但是由于有很多奇怪的逻辑，所以我多写了点。具体用法就是给你的子弹的蓝图加一个组件，然后用就行了。这一块我暂时还没想好，后面可能会根据手感调整。

移动。常用的移动的逻辑我都写进去了。现在的几个stg专用引擎的移动都会有一个问题，就是，在xy模式和va模式里面，往往只有一个，如果你要写一些比较奇怪的动作，你就要去写转换啊什么的。我这儿就给你们直接写好了，里面还留了加速度的。你需要注意的一个细节就是，里面会给你这一帧的偏移量，但是不会自动给你把子弹放到那个位置去，你要自己手动去移动子弹，set location和rotation，好处是，移动之前你可以做任何事情，坏处是，你加了那个组件，然后你发现子弹没动，新手会比较懵逼。

还有一个单独的，是发射子弹的一个什么工具。暂时只写了一次性发射的逻辑，如果你要每多少秒发射一次，那你只有手动去处理时间上的问题。应该是一个delay就完事的。

然后暂时没写的东西。

擦弹没写。重点应该是子弹本身的碰撞也是没有的，意思就是说根本不存在任何的攻击行为。这个暂时都不想补了。

**////自机类////**

我用的英语是player character。

其实就几个事情，移动，然后攻击，b，减速，我额外增加了一个东西，就是攻击的那个按键增加了另外一个按键，是一个toggle，逻辑应该都很简单。

有一些很具体的问题，就是说，玩家在b的时候，这个消息是要发到几个地方的，子弹管理器要去处理消弹的逻辑，然后pawn本身要无敌，然后boss的那个spell card event也需要收到这个消息，所以，b的具体实现，我在考虑要不要搬到主框架去，或者说以后要专门把controller的东西单独拿出来。