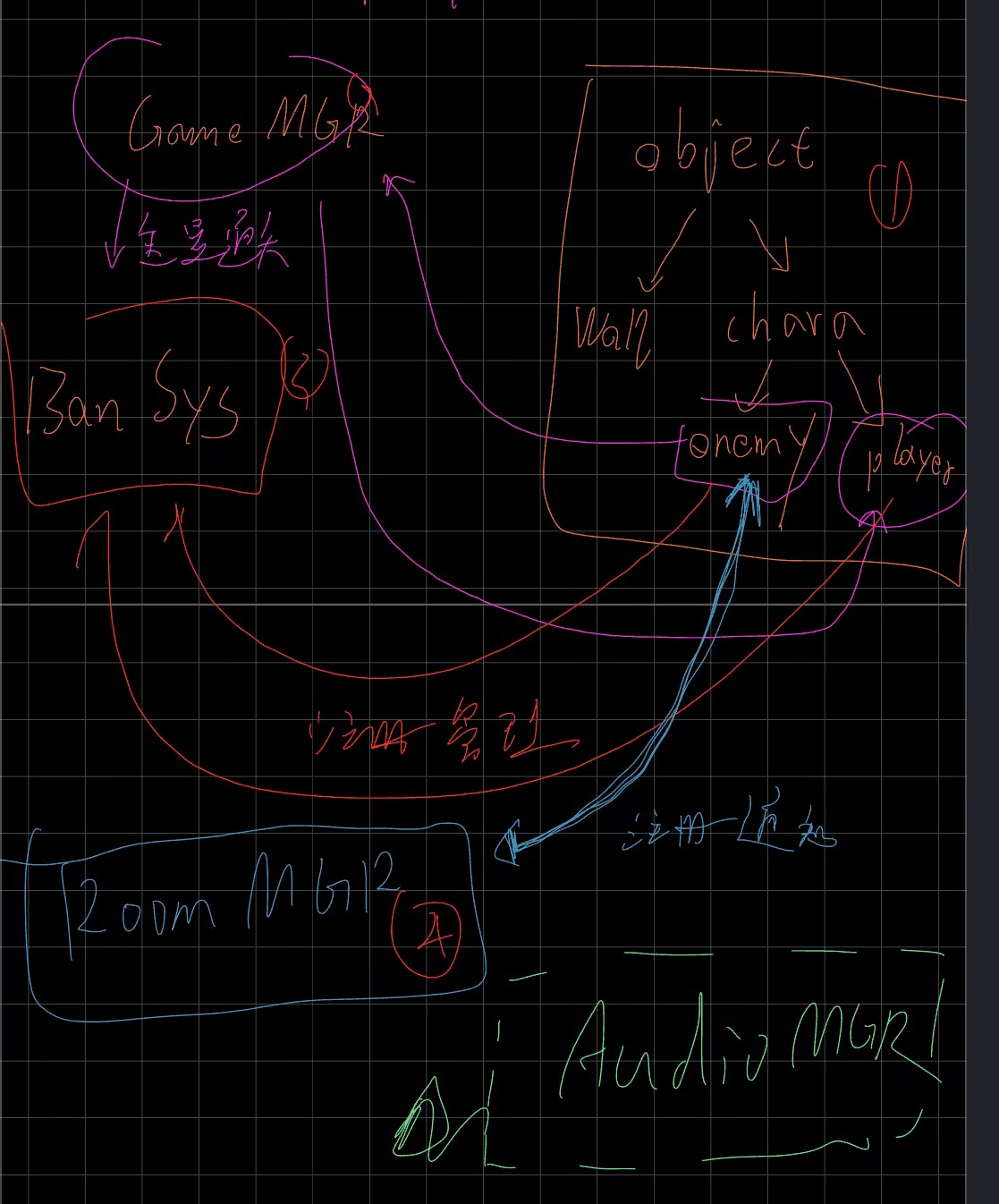
**开发日记**

**Week1 :基础功能**

1. 完成基础功能的规划



对于节奏系统的设计如下,之后会利用插件来进行详细的设计

图示

低可信度描述已自动生成

2.制作和节奏系统相关的Obj的生命周期

因为节奏系统是一个大头，所以一些Obj在一开始的时候就需要完成和Rhy系统的响应的绑定，大概表现为“生成->向RhyMGR注册-> 响应自己的AI -> 响应RhyMGR的AI -> 生命小于1 ->从RhyMGR中remove -> Death函数”

3.设置Layer

其中sortingLayer是为了和layer配合，保持渲染顺序的正确

图形用户界面, 文本, 应用程序

描述已自动生成

而layer和tag是为了区分不同用途的object，其中特别设置Enemy之间不会产生碰撞

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

文本

低可信度描述已自动生成

2.移动和射击模块基本OK了，战斗模块还要做

3.完善基础战斗模块，增加动画

图形用户界面, 日程表

描述已自动生成

4.实现冲刺功能，并且包含急停，和默认冲刺方向

文本

描述已自动生成

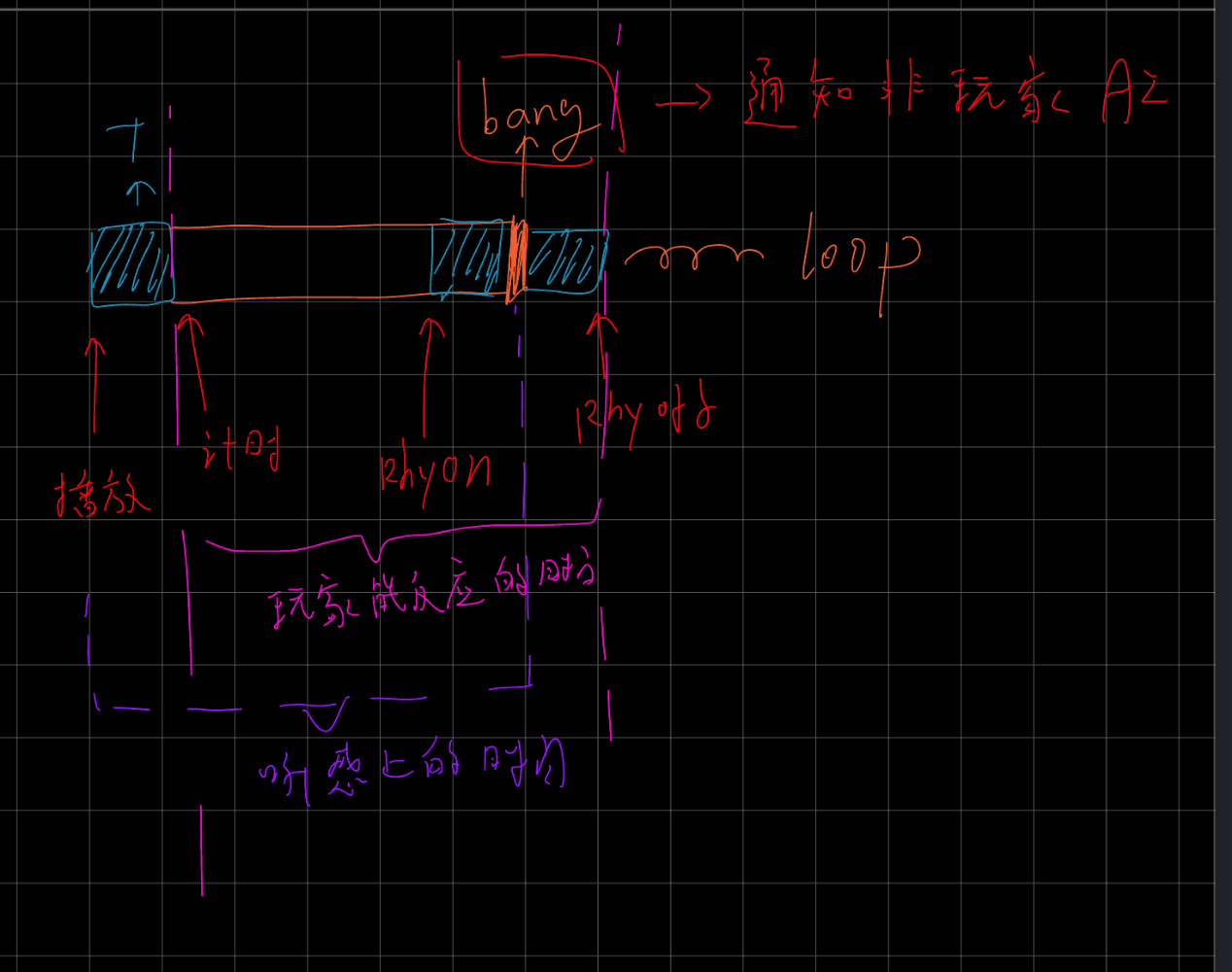
5.尝试接入节奏系统，先使用手动计时

图示

低可信度描述已自动生成

后面优化时会尝试使用**Koreographer**

但是由于玩家响应时间需要有一定延迟，所以设置一个“RhyTolerance”来进行优化，配合UI等项目，实际上代码运行的时间轴是：



**附录**

关于[【Unity】Update，LateUpdate，FixedUpdate 的区别\_fixedupdate()-CSDN博客](https://blog.csdn.net/ZJ_____W/article/details/115178508)

Update : 每一帧执行的时候，立即调用

FixedUpdate : 在所有Update方法调用之后被调用，它适合处理与物理相关的操作，例如刚体碰撞、力的应用等

LateUpdate : 固定更新，默认0.02s调用一次

（时间开发中，碰到以S为单位的判断，就用FixedUpdate最好了）

Unity编辑器中的脚本执行顺序是Awake——>OnEnable——>Start——>FixedUpdate——>yield WaitForFixedUpdata——>OnTrigger——>OnCollision——>OnMouse\*\*（输入事件）——>Update——>yield WaitForSecone——>StartCoroutine——>LateUpdate（游戏逻辑）——>屏幕渲染——>OnGui（GUI渲染）——>OnDisable——>OnDstory。

1.最先被添加到GameObject上的脚本中的Awake、OnEable是最后被调用的，反之最后被添加的则会最先被调用。

2.当每个脚本的Awake、OnEable都被调用完了之后，才会调用各个脚本的Start函数。

Time.deltaTime：以秒计算，前一帧到后一帧所用的时间

Time.fixedDeltaTime：Unity固定时间，为0.02s，可在EditProjectSettings—Time修改

文本

描述已自动生成