一、Player类：

内部类：AttackAction

成员：1.动作伤害，2.使用何种方式攻击，3.攻击特效

1.基础属性模块

1.生命，硬直，移动速度，各状态标志位

2.攻击方式（部位），连招树相关

2.战斗模块

1.目标获取函数（通过Tag）

2.连招树节点定义，连招树的创建（虚函数）

3.接受伤害和制造伤害函数（Get和Make）

4.攻击动画切换与结束（通过状态机）

5.异常状态（眩晕）及恢复

6.控制技能（击飞等）

7.闪避系统

8.攻击控制系统（指令判断，指令缓存，当前执行动作）

3.人物移动模块

1.用户输入移动控制

2.动画附带的自动移动

4.状态机模块（解耦合能力较好）

1.状态初始化

2.状态切换

3.各状态Update函数

5.Update

Not dead

If (no target)

GetTarget();

If(not in abnormal status)

Dodge();

Attack();

Move();

//上面三个状态均互斥

StateMode.Update();//做这个状态下应该做的事

二、怪物类

1、Monster类

1.基本属性与组件

1.同Player中的相关属性

2.寻路系统

3.动作控制模块（MonsterActionManager类）

2.战斗模块

1.目标获取

2.连招序列初始化（虚函数）

3.接受伤害伤害函数（Get）

4.攻击动画切换与结束（通过MonsterActionManager类）

5.异常状态（眩晕，击飞）及恢复

6.控制技能（眩晕等）

3.移动模块

1.通过MoveAgent()前往动作发生地点（虚函数）

2.动画附带的自动移动

3.飞行（虚函数）

4.Update

If(not dead or not in abnormal status)

If (no target)

GetTarget();

ImplementActions();//将某一连招序列发送给MonsterActionManager执行

2.MonsterActionManager类（扩展和解耦能力强， 战斗性能方面较GOAP算法）

1.接受并初始化Monster所有动作序列（攻击）

2.接受动作执行序列

3.动作切换，序列执行结束和动作打断处理（包括循环和非循环动作）

4动作制造伤害函数（Make）

3.ActionBase类

1.动作各属性

名字，是否循环，循环时间，执行范围限制与否，伤害

2.动作重置 //每次执行前重置成员各变量值

3.动作执行（循环与非循环）

1. 场景切换

开始界面（自己画的，一条龙和一个女人（两个Boss）和玩家^-^）和战斗界面的切换

四、Shader

1.常规Shader（Player， Monster）

半兰伯特光照模型和凹凸纹理

2.溶解特效Shader

基于噪声采样的溶解

3.闪避残影特效Shader

缓存并渲染Mesh

4.冲击波特效Shader

后处理特效，基于屏幕像素的世界坐标重建

5.武器蓄力发光特效Shader

通过双Pass的颜色混合