

# 《超新星计划》 末期答辩

客户端开发： 《Mass Fighting》

答辩人：刘梦杰

导 师：钟小武

# 一、自我介绍

➤mengjieliu(刘梦杰)

➤Red客户端战斗组 6.1至今开始实习

➤Mini Game: 《Mass Fighting》

➤导师: 钟小武

➤熟悉Red帧同步流程、工具箱以及Kungfu编辑器相关需求开发

## 二、题目分析—物理和化学

✓化学：游戏世界的一种既定的规则（如元素反应）

✓物理：物理碰撞、物理模拟



## 二、题目分析—化学

✓化学规则设计：技能附着指定元素于Monster

✓附着元素B时，若Monster已存在元素A，则A与B会触发元素反应

	火元素	冰元素	雷元素	水元素	风元素
火元素	——	释放技能“火柱”	对附近敌人造成击 飞伤害	暂时增加攻击力、暴 击率	扩散
冰元素	——	——	降低敌人40%防御力	冻结敌人，无法移动	扩散
雷元素	——	——	——	持续造成伤害	扩散
水元素	——	——	——	——	扩散
风元素	——	——	——	——	——

## 二、题目分析—物理

✓物理规则体现：物理模拟



# 三、项目结果展示

▣技能击中敌人时会附着对应元素在怪物身上

①火元素附着

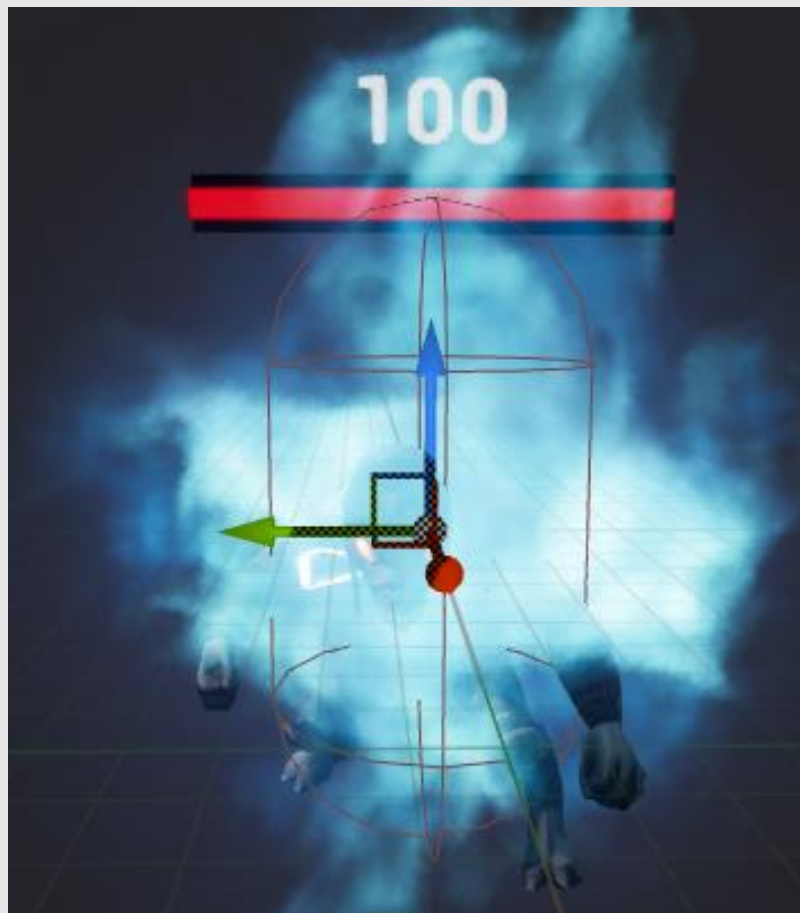


②水元素附着



# 三、项目结果展示

③冰元素附着



④雷元素附着





### 三、项目结果展示

⑤水+雷=感电：周期性造成伤害





### 三、项目结果展示

⑥冰+雷=超导：降低物理防御



⑦水+冰=冻结



### 三、项目结果展示

⑧火+雷=超载：造成范围爆炸伤害



⑨水+冰=融化：生成火柱



### 三、项目结果展示

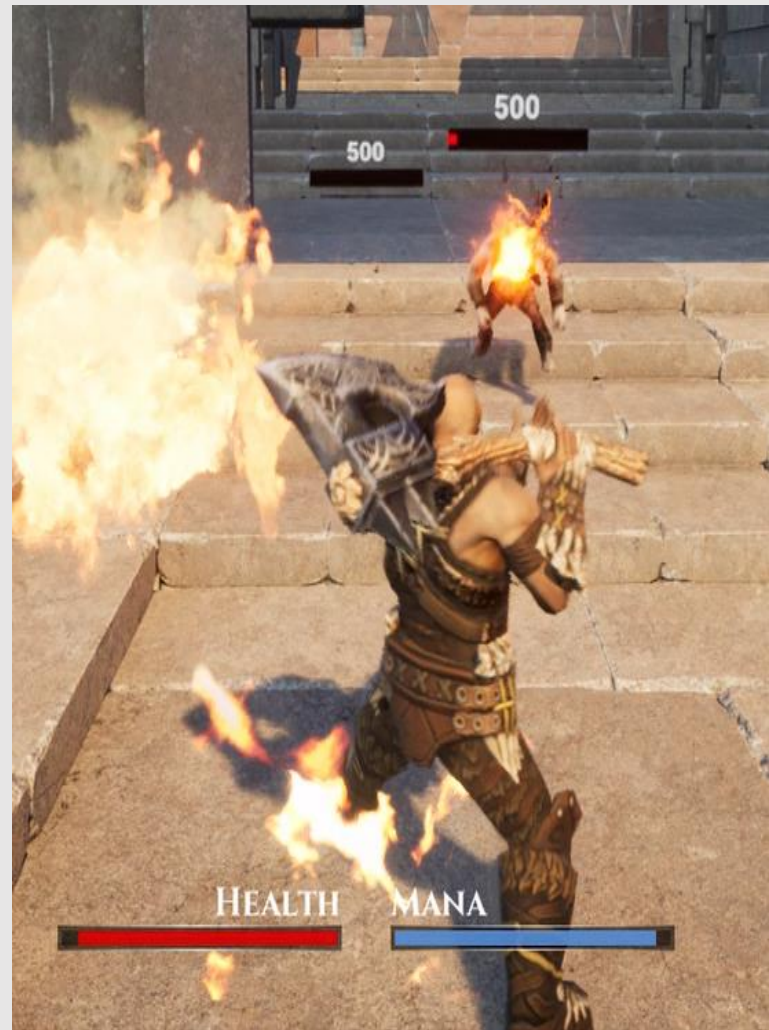
⑩风+其他=扩散：将该元素附着至其他怪物





### 三、项目结果展示

⑪水+火=易伤：短暂时间内提高一倍受到的元素伤害





## 四、完整项目演示

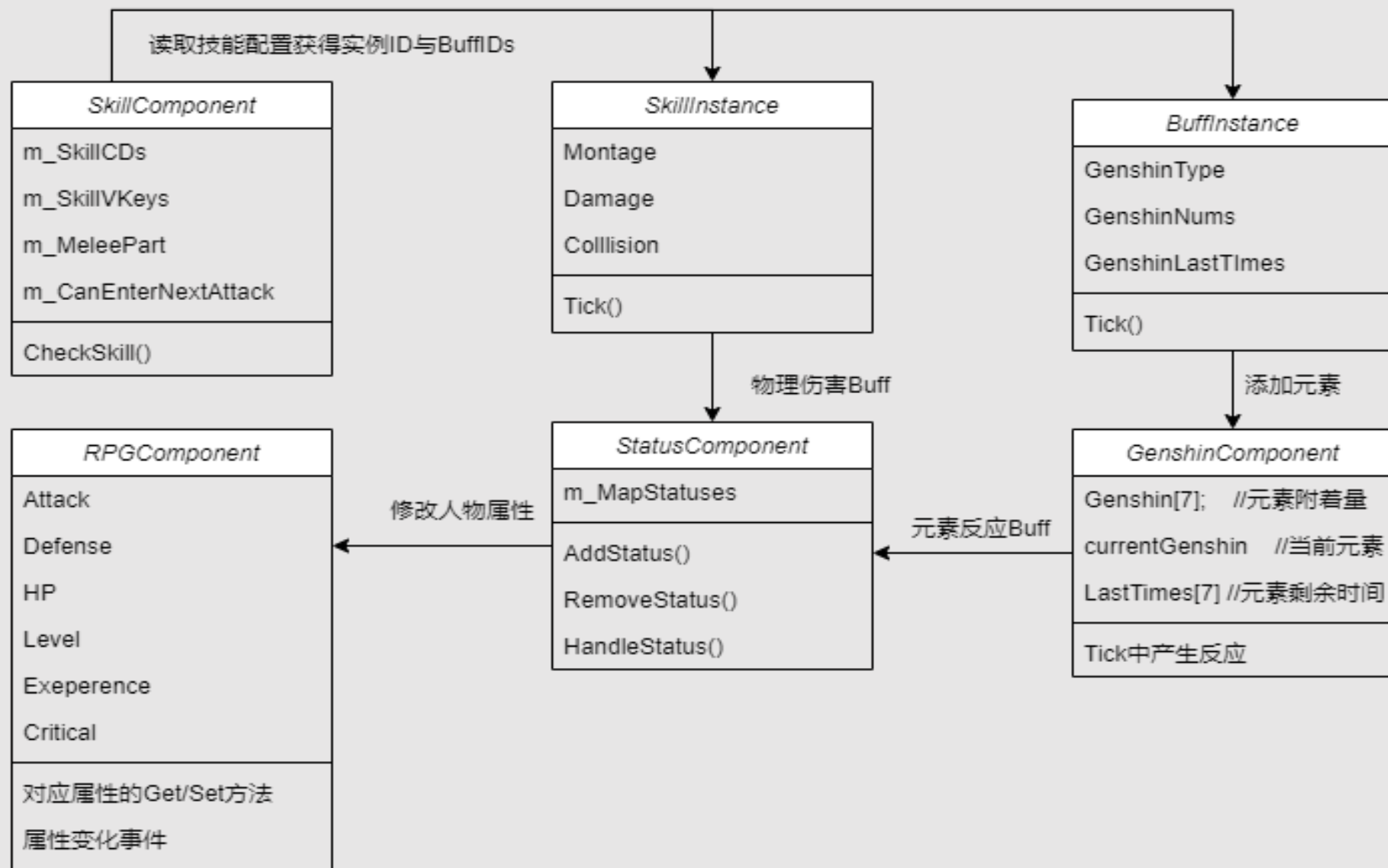


# 五、系统设计—模块设计

- 主要包括四个模块
  - **SkillModule**: 技能释放检测、CD等
  - **GenshinModule**: 元素附着、反应等
  - **BuffModule**: 更新人物当前所拥有的Status
  - **RPGModule**: Buff最终会影响到人物状态



## 五、系统设计—模块交互



# 六、关键技术点设计及实现

## ✓数据驱动

- 简易的技能配置表、Buff配置表
- 一定的拓展性、维护性

## ✓ECS架构

- UE5 MassAI
- ActorComponent/WorldSubsystem  
➔ Fragment/Processor
- Cache高命中率

## 六、关键技术点设计及实现

### ✓数据驱动

```
USTRUCT(BlueprintType)
struct FMJStatusData : public FTableRowBase
{
    GENERATED_BODY()

public:
    FMJStatusData() {}
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "DataTable Status")
    int32 m_StatusID;

    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "DataTable Status")
    FString m_Name;
```

```
USTRUCT(BlueprintType)
struct FMJSkillData : public FTableRowBase
{
    GENERATED_BODY();

public:
    FMJSkillData() {}
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "DataTable Skill")
    int32 m_Vkey;

    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "DataTable Skill")
    FString m_Name;
```

# 六、关键技术点设计及实现

## ✓数据驱动

数据表格 ×											
Q 搜索											
	行命名	M Vkey	M Name	M CD	M Cost	M Break Point	M Damage Ratio	M Genshin	M Anim Montage	M Will Addbuff IDs	M
1	NewRow	100	Melee1	0.000000	0	0	100	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Mon_Melee1.Mi	()	
2	NewRow_0	101	Melee2	0.000000	0	1	110	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Mon_Melee2.Mi	()	
3	NewRow_1	102	Melee3	0.000000	0	2	120	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Mon_Melee3.Mi	()	
4	NewRow_2	103	Roll	0.500000	0	10	0	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	
5	NewRow_3	104	BeHited	0.000000	0	30	0	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	
6	NewRow_4	105	Died	0.000000	0	0	0	None	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Mon_Death.Mo	()	
7	NewRow_5	200	Fire	5.000000	10	30	500	Fire	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	
8	NewRow_6	201	Thunder	4.000000	8	30	300	Thunder	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	
9	NewRow_8	203	Water	6.000000	5	30	350	Water	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	
10	NewRow_9	204	Wind	10.000000	10	30	0	Wind	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	("AddAttack" = True, "AddCrital" = True)	
11	NewRow_10	205	Ice	7.000000	7	30	250	Ice	AnimMontage'/Game/_Fighting/MASSPlayer/Characters/Animations/AN	()	

## 六、关键技术点设计及实现

### ✓数据驱动

数据表格										
搜索										
	行命名	M Status ID	M Name	M Delta Value	M Type	M Duration	M Modify Property	M Exec Interval Time	M Genshin	M Is Consant
1	Fire	100	Fire	2	Genshin	0.000000		0.000000	Fire	False
2	Thunder	102	Thunder	2	Genshin	0.000000		0.000000	Thunder	False
3	Ice	104	Ice	2	Genshin	0.000000		0.000000	Ice	False
4	Water	106	Water	2	Genshin	0.000000		0.000000	Water	False
5	Wind	110	Wind	2	Genshin	0.000000		0.000000	Wind	False
6	Chaodao	200	Chaodao	-10	temporary	10.000000	Defense	0.000000	None	False
7	Zhengfa	201	Zhengfa	100	temporary	10.000000	GenshinRatio	0.000000	None	False
8	Gandian	202	Gandian	-10	Repeat	5.000000	HP	2.000000	None	False
9	Freeze	204	Freeze	-600	temporary	4.000000	Speed	4.000000	None	False

## 六、关键技术点设计及实现

### ✓ECS架构

```
USTRUCT()  
struct FRPGFragment : public FMassFragment  
{  
    GENERATED_BODY()  
  
    //基本属性  
    UPROPERTY()  
    int32 m_HP = 1000;  
  
    UPROPERTY()  
    int32 m_MAXHP = 1000;
```



## 六、关键技术点设计及实现

### ✓ECS架构

```
UCLASS()  
class MASSAI_FIGHTING_API UMJRPGMassProcessor : public UMassProcessor  
{  
    GENERATED_BODY()  
  
    UMJRPGMassProcessor();  
  
protected:  
    virtual void ConfigureQueries() override;  
  
    virtual void Execute(UMassEntitySubsystem& EntitySubsystem, FMassExecutionContext&
```

## 六、关键技术点设计及实现

### ✓ECS架构

```
void UMJRPGMassProcessor::Execute(UMassEntitySubsystem& EntitySubsystem, FMassExecutionContext& Context)
{
    EntityQuery.ForEachEntityChunk(EntitySubsystem, Context, ([this](FMassExecutionContext& Context)
    {
        TArrayView<FRPGFragment> RPGList = Context.GetMutableFragmentView<FRPGFragment>();
        TArrayView<FSkillFragment> SkillList = Context.GetMutableFragmentView<FSkillFragment>();
        const TArrayView<FTransformFragment> LocationList = Context.GetMutableFragmentView<FTransformFragment>();

        const TConstArrayView<FMassEntityHandle> HandleList = Context.GetEntities();

        const float WorldDeltaTime = Context.GetDeltaTimeSeconds();
        UMJDeleSubsystem* delesub = GetWorld()->GetSubsystem<UMJDeleSubsystem>();

        for (int32 EntityIndex = 0; EntityIndex < Context.GetNumEntities(); ++EntityIndex)
        {
```

# 附录

## ●资源引用

- 地图：古代山谷
- 人物骨骼动画：ARPG
- 部分UI：ARPG
- 技能粒子特效

## ●Demo引用：无