

# 技能逻辑分析

---

描述技能类型与逻辑时序

## 技能类型

---

常见的技能类型：

### 近战

#### 1. 近战单段普攻过程

挥刀（攻击动作+特效）-> 固定时间后触发命中判定-> 对象受击（受击动作+特效）-> 伤害计算

#### 2. 近战多段普攻过程

重复多遍1的过程

#### 3. 近战技能

施法前摇（施法动作）-> 固定时间后触发-> 命中判定-> 技能效果处理（受击或加buff）

### 远程

#### 1. 远程普攻过程

施法前摇（施法动作）-> 固定时间后触发子弹生成-> 飞向目标-> 对象受击（受击动作+特效）-> 伤害计算

#### 2. 远程技能过程

施法前摇（施法动作）-> 固定时间后触发-> 子弹生成或直接目标点判定-> 命中（命中/受击特效+动作）-> 技能效果处理

## 技能时序

---

技能时序可统一描述为：

1. 施法前摇（施法动作+特效）
2. 固定时间（或为0）后触发下一步
3. 生成“子弹”（如果有，没有则直接执行第4步），飞向目标
  - 有飞行轨迹：直线/曲线
    - 固定飞行时间
    - 固定飞行速度
  - 无飞行轨迹
4. 命中判定
  - 单体判定（锁定目标）
    - 命中概率
  - 范围判定（锁定目标，或无锁定目标）
    - 个数限制
    - 命中概率

## 5. 效果执行

# 技能编辑器Flux

---

需要支持的编辑模式：

- ☒ 物件的出现与消失
- ☒ Animation播放
- ☒ 物件移动/旋转/缩放
- ☒ 粒子的播放
- ☒ 镜头抖动
- ☒ 物件的生成
- ☒ 声音的播放
- ☒ 判定逻辑帧
- ☒ 投射逻辑帧

需要导出的数据：

1. 判定的时间线
2. 子弹的飞行轨迹

## 例子

---

	流程	代码	系统
1	点击技能按钮	function BattleRequests:CastSkill(id)	Lua
2	逻辑层输入处理系统接收请求	CastSkill::OnProcess(CastSkillRequest request, CastSkillRespond respond)	InputSystem(L)
3	尝试使用指定技能进行施法	SkillUtils.TryCastWithAbility(Contexts, entity, ability, out var target);	SkillUtils(L)
	3.1 校验施法状态	SkillCastResult ValidateCastContext(LogicContexts contexts, LogicThingEntity thingEntity, SkillConfData ability)	SkillUtils(L)
	3.2 施法对象的查找, 排序, 以及过滤	QueryEntities/Filter/Sort	SkillUtils(L)
4	对应实体进入施法状态	thingEntity.Cast(context)	ThingSystem(L)
	4.1 视图层消息系统处理施法消息	OnEnterCast::OnProcess(ThingEnterCastMessage message)	InputSystem(V)
	4.2 创建技能视图实体	AddSkillCastViewSystem::CreateSkillView(ViewSkillEntity skillEntity)	SkillSystem(V)
5	创建施法抽象实体	Contexts.logicSkill.CastSkill(Contexts,thingEntity, stateContext.TargetId, stateContext.Ability);	SkillSystem(L)
6	读取施法序列数据	CastSkillSystem::ReadAndProcessSkillSequence(LogicSkillEntity skillEntity)	SkillSystem(L)
7	等待时间到达, 触发执行事件 (Judge/Shoot)	ProcessSkillFluxEventSystem::ProcessSkillEvent(LogicSkillEntity skillEntity, SkillFluxEventContext fluxEvent)	SkillSystem(L)
		FluxJudgeEventProcessor::Process(ulong casterId, ulong targetId, SkillFluxEventContext fluxEventContext)	SkillSystem(L)
	7.1 判定生效目标的查找, 排序, 以及过滤(范围性技能)	FluxJudgeEventProcessor::CollectAffectedTargets(SkillFluxEventContext fluxEvent, ActiveSkillEffectData effectData, ulong casterId, ulong targetId, ref List realTargets)	SkillSystem(L)
	7.1 视图层消息系统处理命中消息	OnSkillHit::OnProcess(SkillJudgeHitMessage message)	InputSystem(V)
	7.2 创建命中与受击表现	CreateSkillHitEffect(SkillJudgeHitMessage message, ulong targetId) CreateAvatarHurtEffect(ulong casterId, ulong targetId)	InputSystem(V)
8	创建效果实体	var effectEntities = Contexts.AddEffects( casterId, realTargets, effectData.EffectData, true, EffectSource.Skill, userData);	SkillSystem(L) EffectExtensions(L)
9	处理具体效果	AddEffectSystem::ProcessEffect(LogicEffectEntity effectEntity, EffectData effectData)	EffectSystem(L)
	9.1 根据子类型创建不能的效果处理器	EffectAdderFactory.CreateProcessor(Contexts, effectData)	EffectSystem(L)