1. **ComponentReceiverActors**
   1. **概述**
      1. 系统层面非内容插件。
      2. 实现模块化的基础。
      3. 提供一套Actors，使其在启动时动态加载组件的功能。
   2. **C++层**
      1. GameModeBase
      2. GameMode
      3. GameStateBase
      4. GameState
      5. PlayerState
      6. Pawn
      7. Character
      8. PlayerController
      9. AIController
   3. **蓝图层**
2. **PawnInput**
   1. **概述**
      1. 系统层面非内容插件。
      2. 依赖EnhanceInput，提供IM加载卸载的方案。
      3. 向高层提供IA与具体行为绑定解绑的接口。
   2. **C++层**
      1. PawnInputModComponent
   3. **蓝图层**
3. **MovementInput**
   1. **概述**
      1. 面向玩家的内容插件。
      2. 依赖PawnInput，实现IA对角色移动性能的具体控制方案。
   2. **C++层**
      1. CharacterMovementInputComponent
         1. 继承自PawnInputModComponent。
         2. 实现调整视野和基本移动操作。
   3. **蓝图层**
      1. BP\_MovementInputComponent
         1. 保存IM、IA的参考。
      2. 功能表
         1. 对Character添加BP\_MovementInputComponent。
4. **AbilitiesInput**
   1. **概述**
      1. 面向玩家的内容插件。
      2. 依赖GAS、PawnInput，实现IA对GA释放的具体控制方案（在按下/释放时，激活/去激活能力。）。
      3. 定义Actor通过配表获取属性、技能的具体行为。
   2. **C++层**
      1. GrantAbilitySystemComponent
         1. 继承自AbilitySystemComponent。
         2. 具有一套默认能力和属性，在Actor初始化时这些能力和属性会被赋予携带此组件的Actor。
      2. AbilitiesInputComponent
         1. 继承自PawnInputModComponent.
         2. 实现IA的按下和释放操作到GA的按下和释放方法的代理绑定解绑。
      3. GameFeatureAction\_AddAbilities
         1. 定义能力配表方式：能力GA可与一个操作IA绑定。
      4. GameFeatureAction\_AddAttributes
         1. 定义属性配表方式：属性AS可以在初始化时从一张元数据表DT读取。
   3. **蓝图层**
5. **GameFeatureExpansions**
   1. **概述**
      1. 是对引擎内置插件GameFeature的拓展。
      2. 定义了数据配置表的元和加载方式。
   2. **C++层**
      1. GameFeatureAction\_AddSpawnActors
   3. **蓝图层**
6. **BattleFeature**
   1. **概述**
      1. 依赖GAS，负责具体战斗功能的实现。
      2. 参考文档《连招系统设计方案》
   2. **C++层**
      1. HealthAttributeSet
      2. MovementAttributeSet
      3. GameplayAbility\_Operate
      4. GameplayTask\_PlayMontageWithTag
   3. **蓝图层**
      1. GA\_Run
      2. GA\_MainHand
      3. GA\_OffHand
7. **BattleAI**
   1. **概述**