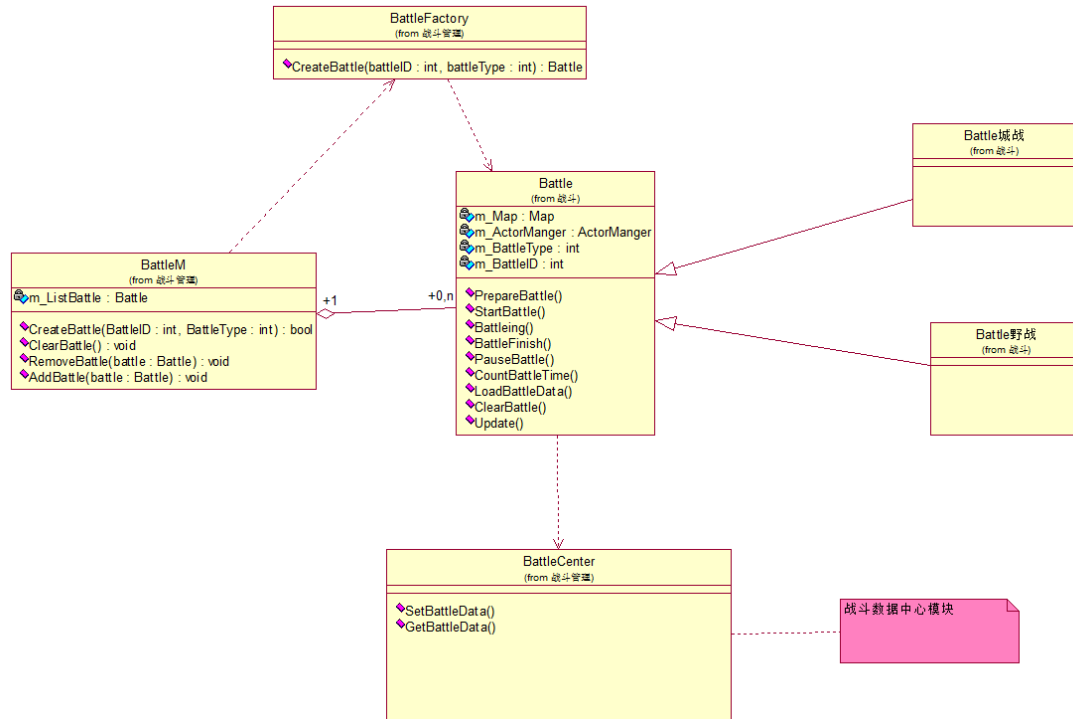


战斗框架设计

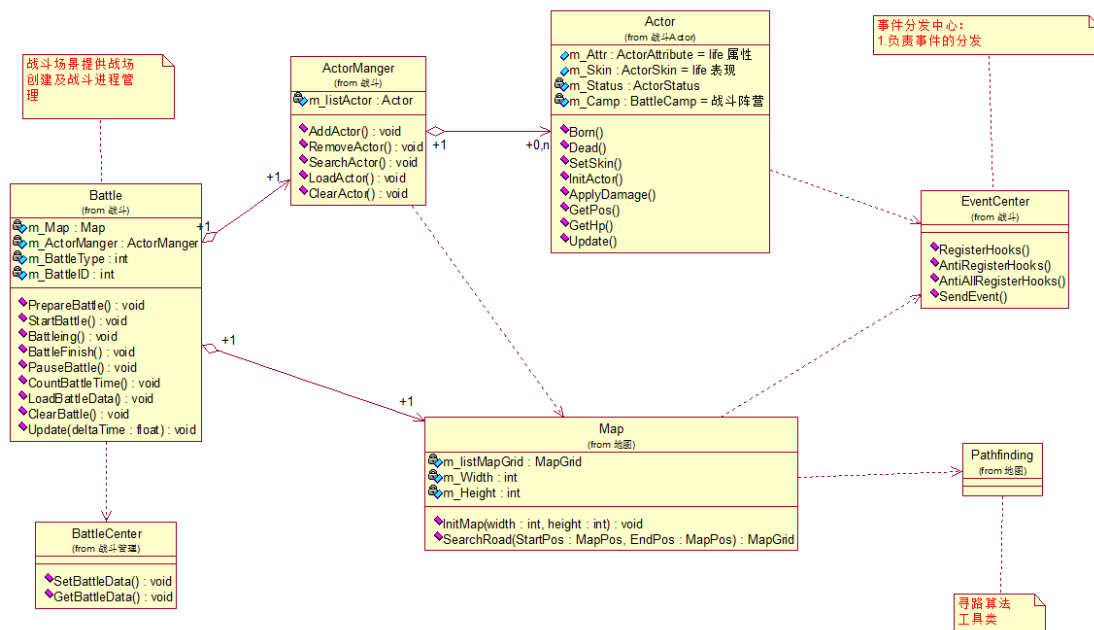
IGG RD3 朱竹林

一.类图

1.战斗管理模块类图设计



2.战斗模块类图设计



3.战斗角色 Actor 设计

```

classDiagram
    class Actor {
        +ActorAttribute life
        +ActorSkin skin
        +ActorStatus status
        +BattleCamp camp
        +Born()
        +Dead()
        +SetSkin()
        +InitActor()
        +ApplyDamage()
        +GetPos()
        +GetHp()
        +Update()
    }
    class ActorAI {
        +Actor parent
        +ActorRadarAI radarAI
        +ActorMove move
        +AttackAI attackAI
        +bool radarEnable
        +bool moveEnable
        +bool attackEnable
        +ProcessEvent(event : Event) void
        +Update(deltaTime : float) void
    }
    class ActorMove {
        +Actor parent
        +MovePath listPath
        +Vector3 velocity
        +CheckNextPosWeight() void
        +Wait() void
        +FindRoad()
        +CalcVelocity() void
        +ClearRoad() void
        +Init() void
        +Update(deltaTime : float) void
    }
    class ActorRadarAI {
        +Actor parent
        +Actor radarTarget
        +FindRadarTarget() Actor
        +Init() void
    }
    class AttackAI {
        +Skill listSkill
        +Skill readySkill
        +Skill bigSkill
        +Actor parent
        +Actor listAttackTarget
        +SearchAttackTraget()
        +RealseSkill()
        +Update()
        +InitSkillData()
        +RealseStatus()
        +CalcSkillDamage()
    }
    Actor "1" --> "1" ActorAI
    ActorAI "1" --> "1" ActorMove
    ActorAI "1" --> "1" ActorRadarAI
    ActorAI "1" --> "1" AttackAI
  
```

The diagram illustrates the Actor class hierarchy and its associations. The Actor class is the base class, and ActorAI, ActorMove, ActorRadarAI, and AttackAI are its subclasses. ActorAI is associated with ActorMove, ActorRadarAI, and AttackAI. ActorAI also has a self-association.

Actor Class (from 战斗Actor)

- Attributes:
 - m_Attr : ActorAttribute = life 属性
 - m_Skin : ActorSkin = life 表现
 - m_Status : ActorStatus
 - m_Camp : BattleCamp = 战斗阵营
- Operations:
 - Born()
 - Dead()
 - SetSkin()
 - InitActor()
 - ApplyDamage()
 - GetPos()
 - GetHp()
 - Update()

ActorAI Class (from AI)

- Attributes:
 - m_parent : Actor
 - m_RadarAI : ActorRadarAI
 - m_Move : ActorMove
 - m_AttackAI : AttackAI
 - m_RadarEnable : bool
 - m_MoveEnable : bool
 - m_AttackEnable : bool
- Operations:
 - ProcessEvent(event : Event) : void
 - Update(deltaTime : float) : void

ActorMove Class (from 普通AI)

- Attributes:
 - m_parent : Actor
 - m_listPath : MovePos
 - m_velocity : Vector3
- Operations:
 - CheckNextPosWeight() : void
 - Wait() : void
 - FindRoad()
 - CalcVelocity() : void
 - ClearRoad() : void
 - Init() : void
 - Update(deltaTime : float) : void

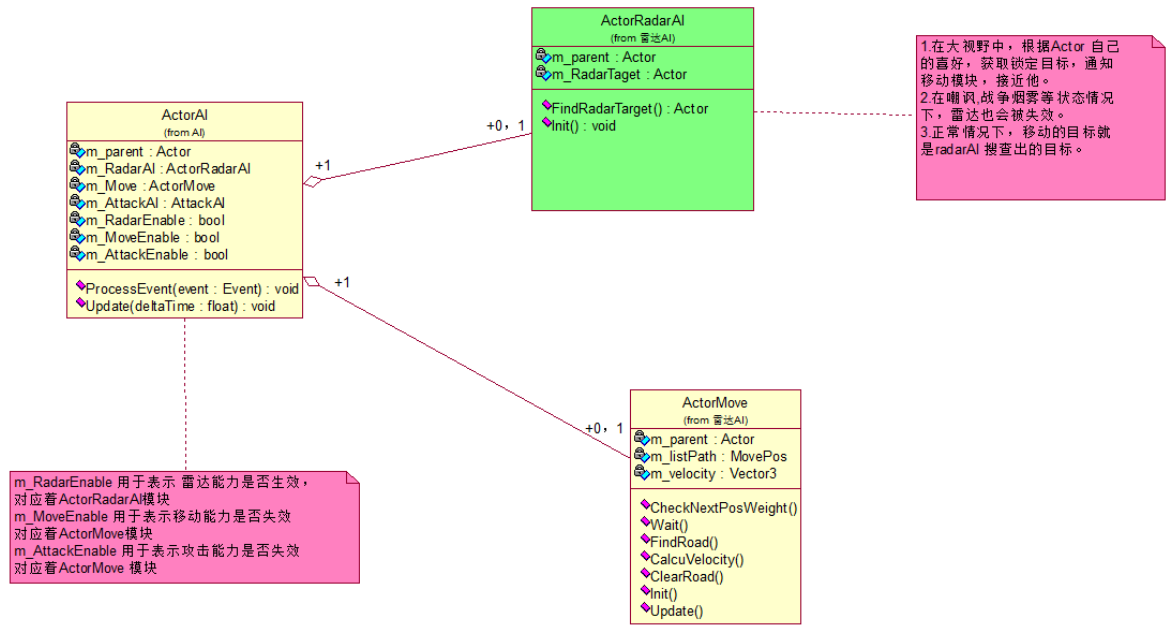
ActorRadarAI Class (from 普通AI)

- Attributes:
 - m_parent : Actor
 - m_RadarTarget : Actor
- Operations:
 - FindRadarTarget() : Actor
 - Init() : void

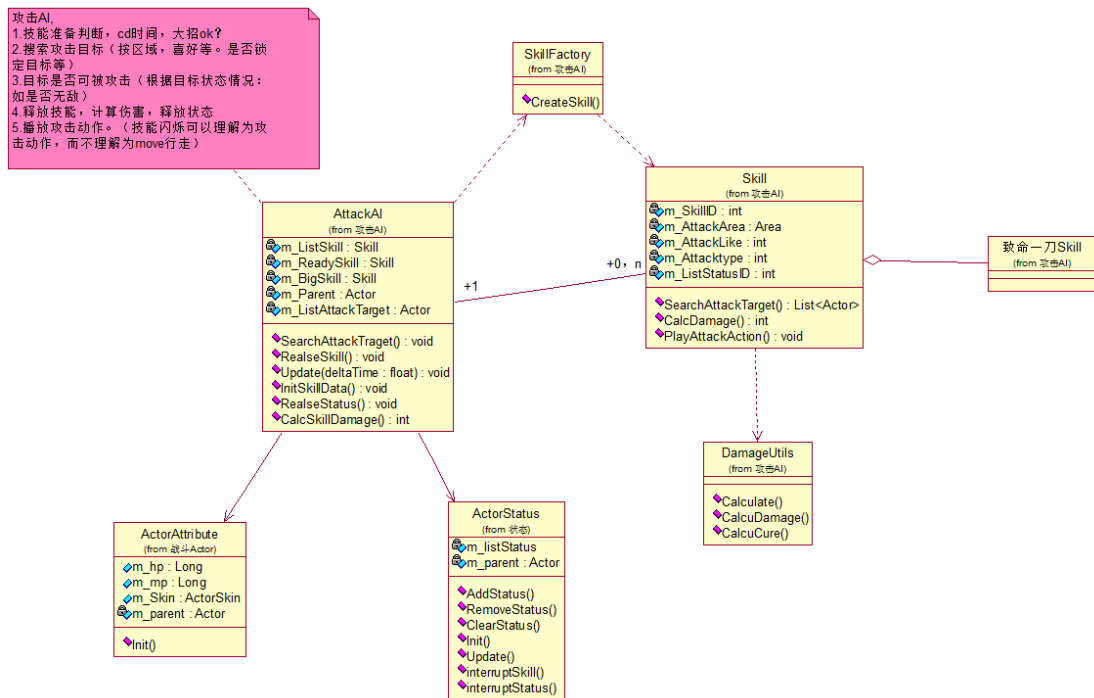
AttackAI Class (from 攻击AI)

- Attributes:
 - m_ListSkill : Skill
 - m_ReadySkill : Skill
 - m_BigSkill : Skill
 - m_Parent : Actor
 - m_ListAttackTarget : Actor
- Operations:
 - SearchAttackTraget()
 - RealseSkill()
 - Update()
 - InitSkillData()
 - RealseStatus()
 - CalcSkillDamage()

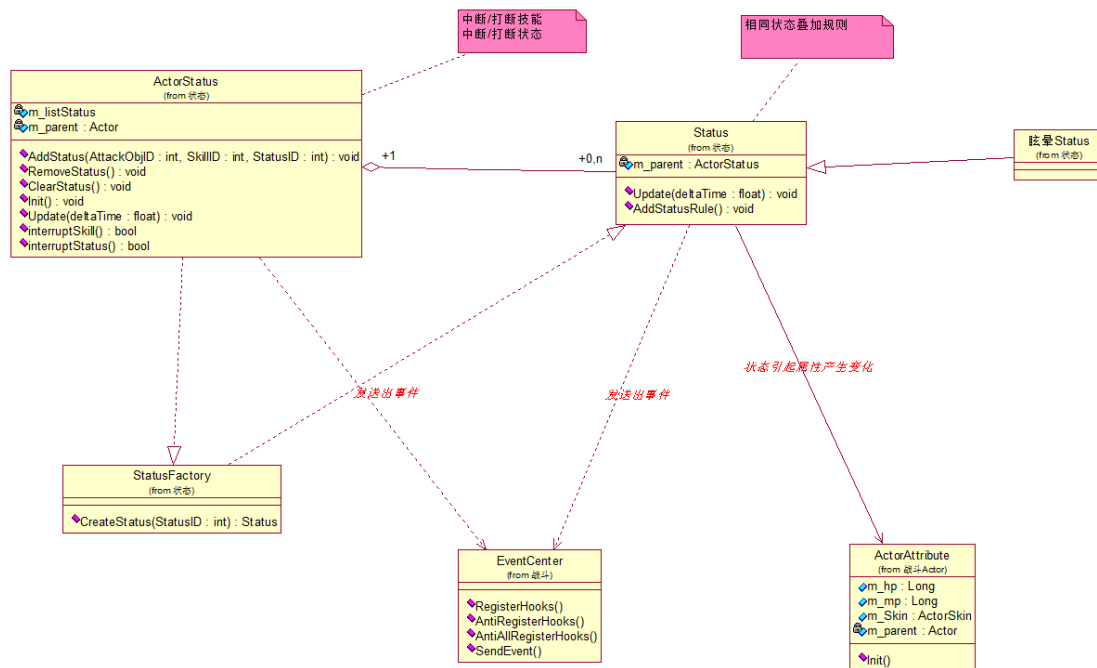
5.Actor 雷达 AI 类图设计



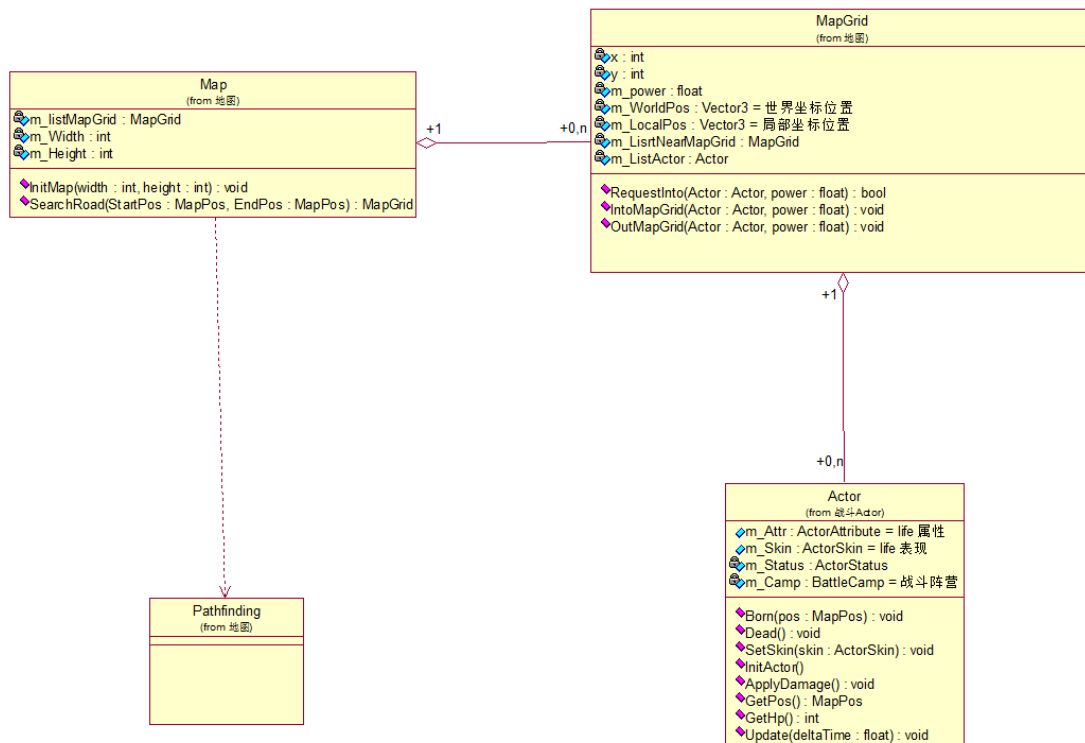
6.Actor 攻击 AI 类图设计



7. Actor 状态类图设计

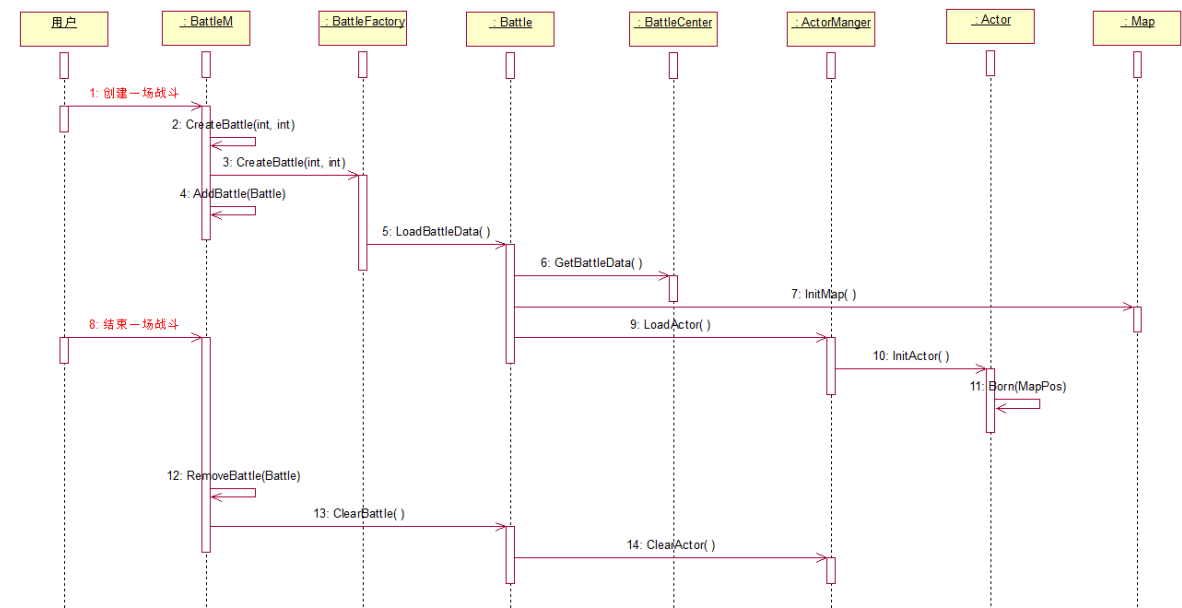


8. 地图 类图设计

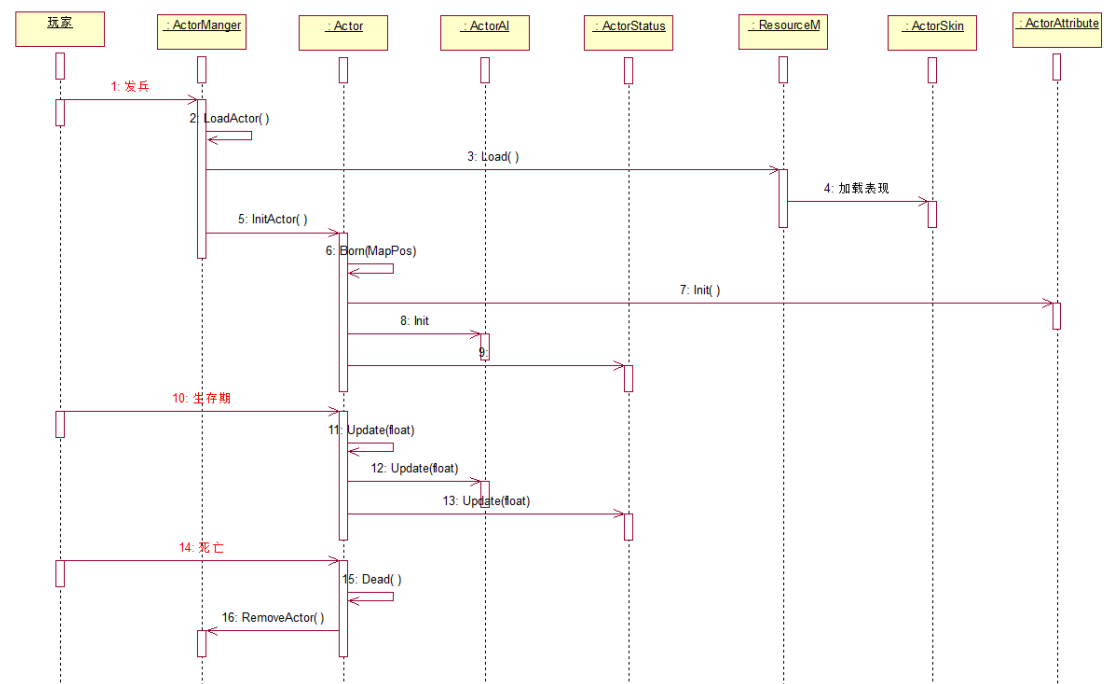


二.时序图

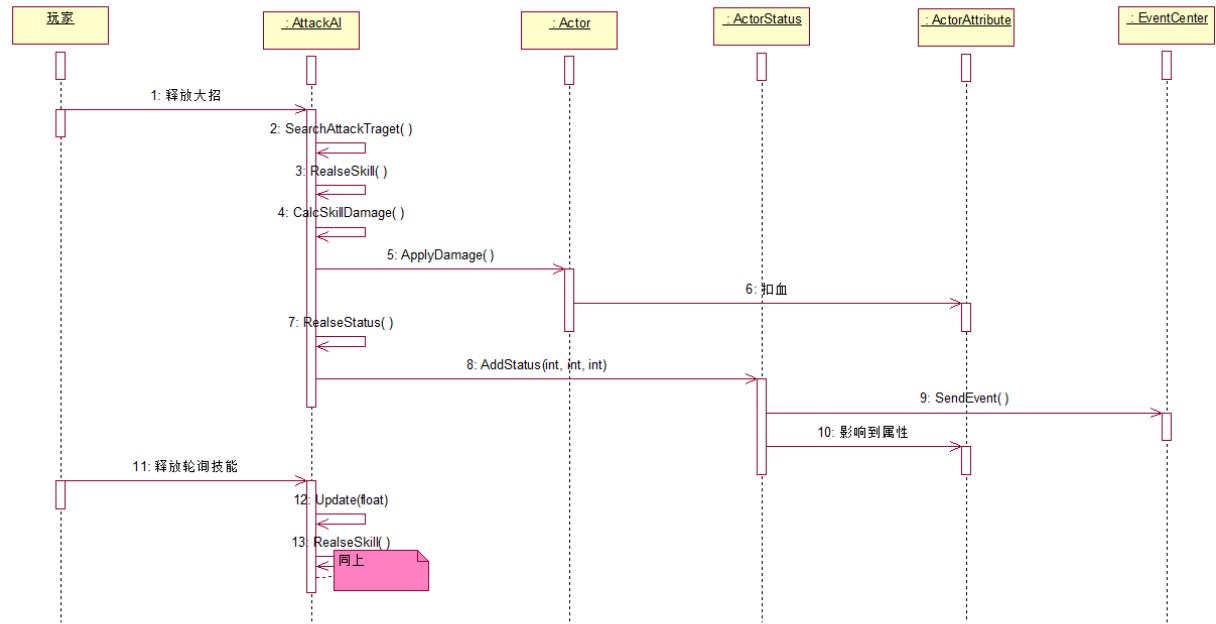
1.构建战斗时序图。



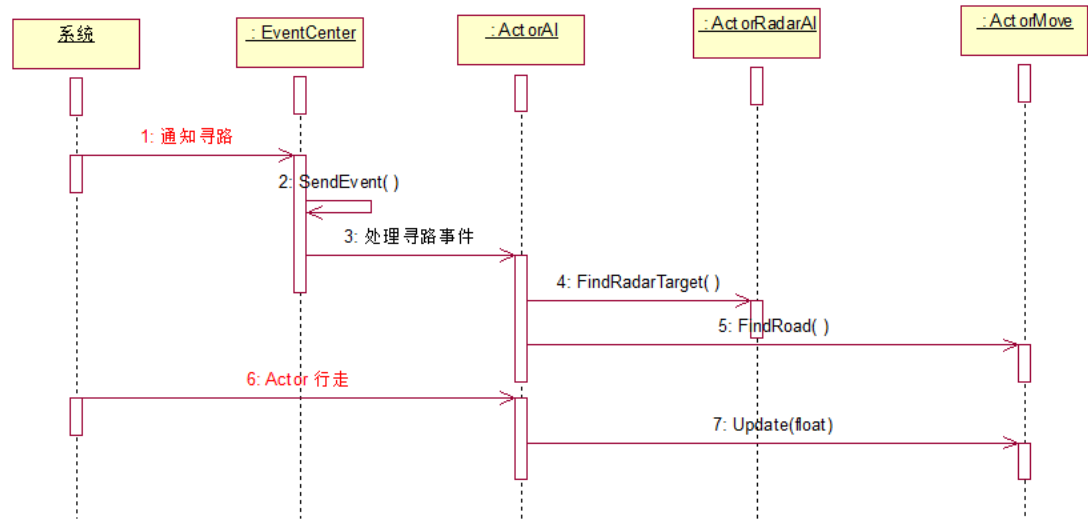
2.角色 Actor 生命周期时序图



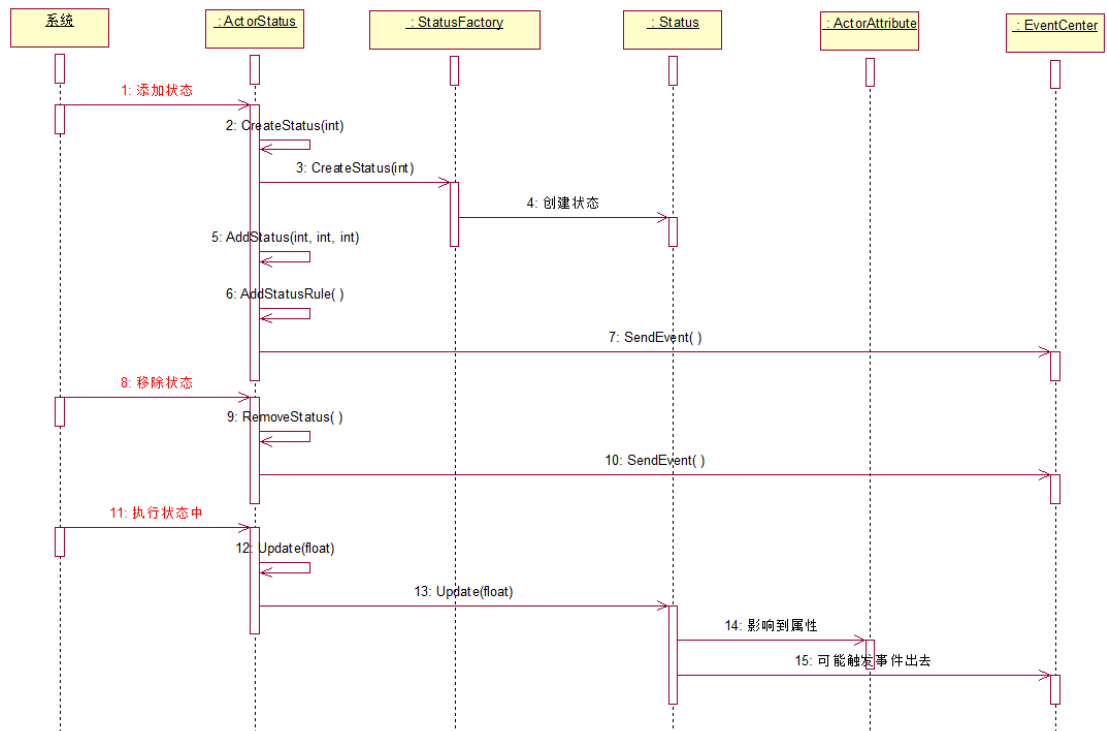
3. 角色 Actor 攻击时序图



4. 角色 Actor 移动/运动时序图



5. 角色状态时序图



三.状态图

1.角色生命周期状态图

