主角攻击妖怪,妖怪要有一些可能性的行为反应——有限状态机

**《状态转换表》**

|  |  |  |  |
| --- | --- | --- | --- |
| **1待机** | **3.1** | **1生命为0** | **死亡** |
| **3.2** | **2发现目标** | **追逐** |
| **2死亡** |  |  |  |
| **3追逐** |  | **生命为0** | **死亡** |
| **3.3** | **3目标进入攻击范围** | **攻击** |
|  | **4丢失玩家** | **转为默认状态** |
| **4攻击** |  | **生命为0** | **死亡** |
| **3.4作业** | **8玩家离开攻击范围** | **追逐** |
|  | **5打死目标** | **转为默认状态** |
| **5巡逻** |  | **生命为0** | **死亡** |
|  | **发现目标** | **追逐** |
| **3.5** | **9完成巡逻** | **待机** |

阶段1：待机》【生命为0】》死亡

阶段2：待机》【发现目标】》追逐

阶段3：追逐》【目标进入攻击范围】》攻击

阶段4：攻击》【玩家离开攻击范围】》追逐

阶段5：巡逻》【完成巡逻】》待机

### 分析：

#### 分析思路

状态与条件的数量及种类不断变化，所以将变化的部分单独封装，抽象状态类及条件类，未来有新状态及新条件时，增加新的状态或条件派生类即可，这样设计是满足开-闭原则的。修改成本最小。

状态的切换与发生的条件相关，所以由状态对象来管理本状态相关联的条件对象，在本状态做为当前状态时，执行所有条件对象的检测（Reason）并执行当前状态提供的行为（Action），如果有条件发生时，则调用状态机来改变当前状态（ChangeActiveState）。

状态机是一个包含AI所需所有状态的容器，负责状态对象的初始化配置，负责状态的切换；状态机实时执行（Update）当前状态的行为(currentState.Action)及条件检测(currentState.Reason)。

1. 各个状态要独立，各自封装一个类,抽象一个状态父类

2. 各个条件要独立，各自封装一个类，抽象一个条件父类

状态可能会有增加或减少 --加新状态类

转换条件可能会有增加或减少或修改-加新条件类

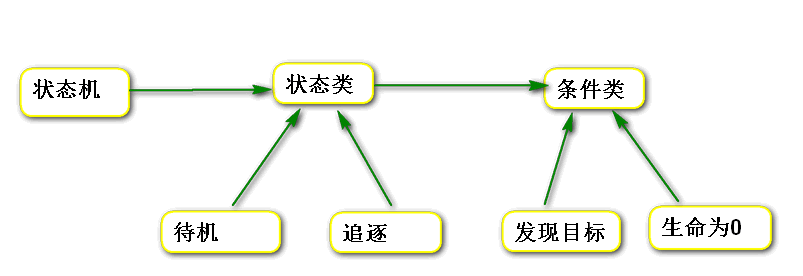
3. 条件与状态是有关系,一个状态中可能拥有多个条件对象.

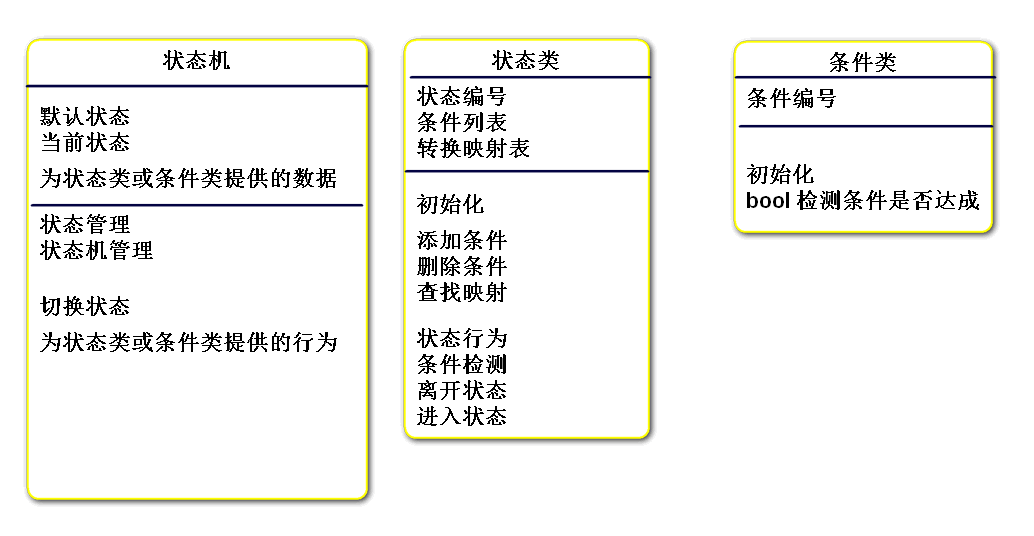
每个状态中的每个条件都对应一个输出结果,每个状态都有一个条件转换状态映射表

4. 同样转换条件可能在不同状态下输出的状态可能会不同,具体的转换过程由谁来做?

定义一个状态机，负责管理AI所需的所有状态.并且协调和组织转换的过程

#### 有限状态机类结构

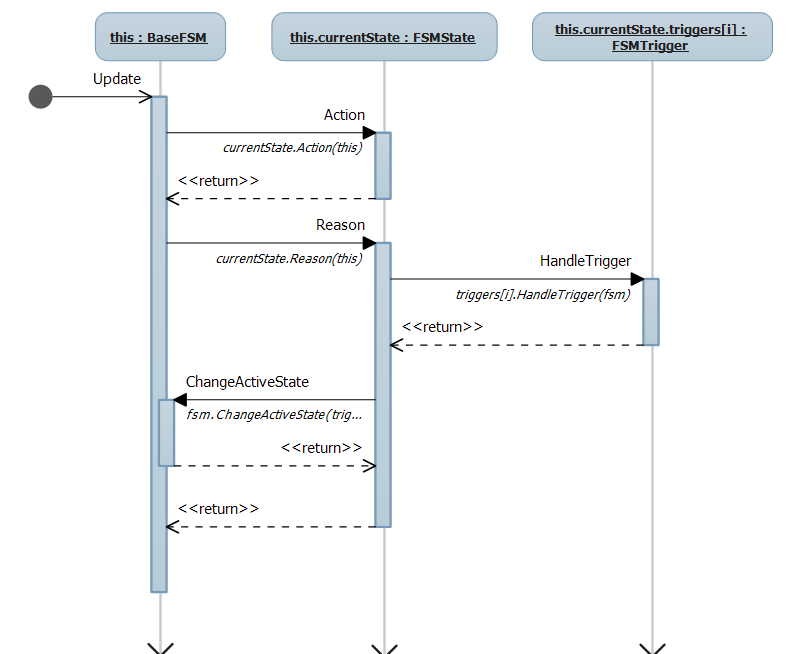




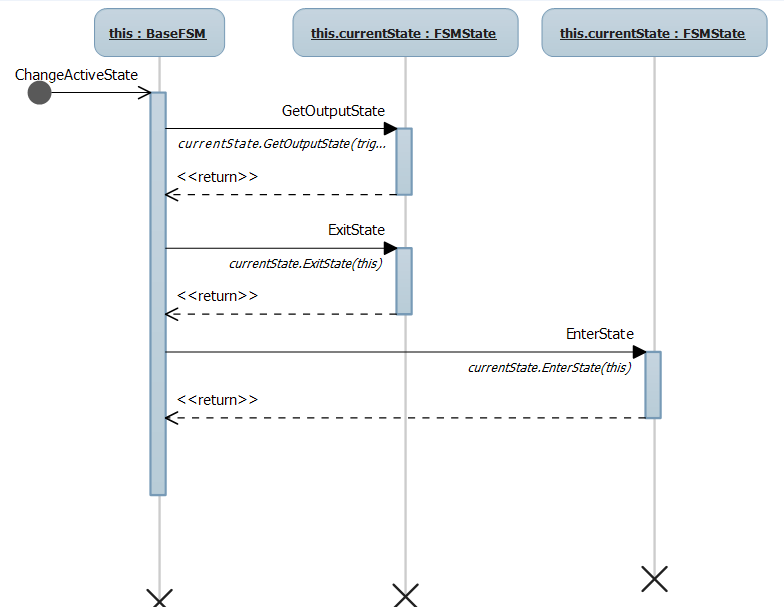
#### 工作原理

当前状态工作时，如果检测到条件达成，则退出当前状态，进入新状态。

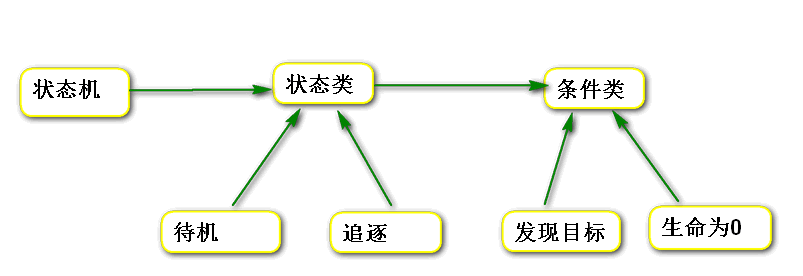
状态机 控制状态的切换！

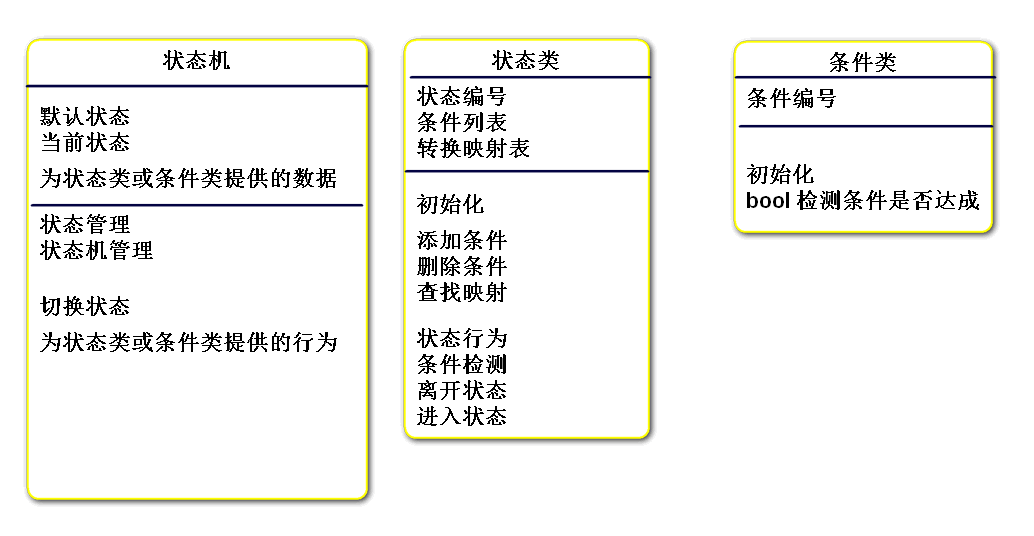


**状态机工作原理**



### 设计类图





#### 类图1：状态编号 FSMStateID

|  |  |  |
| --- | --- | --- |
|  | **状态编号 FSMStateID** | |
| **1** | **无** | **None** |
| **2** | **待机** | **Idle** |
| **3** | **死亡** | **Dead** |
| **4** | **追逐** | **Pursuit** |
| **5** | **攻击** | **Attacking** |
| **6** | **默认** | **Default** |
| **7** | **巡逻** | **Patrolling** |

#### 类图2：状态转换条件： FSMTriggerID

|  |  |  |
| --- | --- | --- |
|  | **转换条件 FSMTriggerID** | |
| **1** | **生命为0** | **NoHealth** |
| **2** | **发现目标** | **SawPlayer** |
| **3** | **目标进入攻击范围** | **ReachPlayer** |
| **4** | **丢失玩家** | **LosePlayer** |
| **5** | **完成巡逻** | **CompletePatrol** |
| **6** | **打死目标** | **KilledPlayer** |
| **7** | **目标不在攻击范围**  **玩家离开攻击范围** | **WithOutAttackRange** |

#### 类图3：动画参数类AnimationParams

[Serializable]

public class AnimationParameter

{

public string Idle = "idle";

public string Dead = "dead";

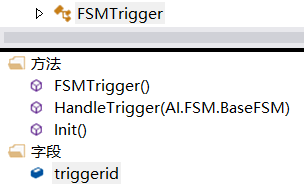
public string Run = "run";

public string Walk = "walk";

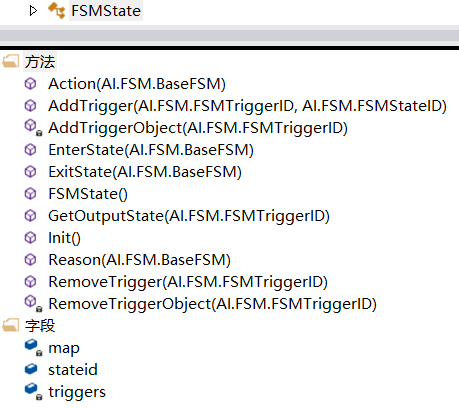
public string Attack = "attack";

}

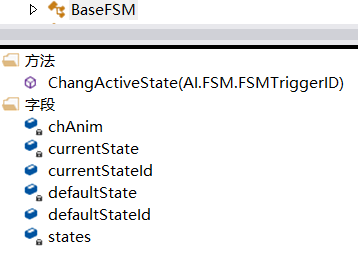
#### 类图4：条件抽象类



#### 类图5：状态抽象类

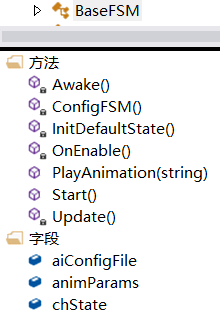


#### 类图6：状态机 BaseFSM

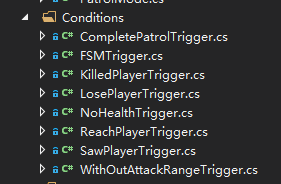


#### 类图7：BaseFSM2.0

给状态机BaseFSM增加以下功能



**状态类脚本**



**条件类脚本**

