

# 有限状态机在模拟演练系统中的应用<sup>①</sup>

张博伦<sup>1,2</sup>, 王 宁<sup>2</sup>, 马慧敏<sup>3</sup>

<sup>1</sup>(中国科学院研究生院, 北京 100049)

<sup>2</sup>(中国科学院沈阳计算技术研究所, 沈阳 110168)

<sup>3</sup>(东软集团股份有限公司, 沈阳 110179)

**摘 要:** 在研究了有限状态机的基本原理和在游戏开发中的应用, 并结合应急模拟演练系统的实际需求, 通过在 Delta3D 引擎中实现一个有限状态机控制器组件解决系统角色状态控制的问题. 首先介绍了有限状态机的基本原理和 Delta3D 引擎架构的特点, 设计了一个有限状态机的例子并详细介绍了有限状态机控制器组件的总体结构、基本功能和具体实现方式.

**关键词:** 有限状态机; Delta3D 引擎; 组件

## Application of the Finite State Machine to Simulation Training System

ZHANG Bo-Lun<sup>1,2</sup>, WANG Ning<sup>2</sup>, MA Hui-Min<sup>3</sup>

<sup>1</sup>(Graduate School, Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology of Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(Neusoft group Co., Ltd, Shenyang 110179, China)

**Abstract:** In the study of the basic theory of the finite state machine and the application of the game development, combined with the actual needs of the emergency simulation training system, solve the system role of state controlling problem with a finite state machine controller components in Delta3D engines. This paper first introduces the basic theory of the finite state machine and the characteristics of the Delta3D engine architecture, designs a finite state machine example and describes in detail the finite state machine controller components of the overall structure, the basic functions and specific implementing way.

**Key words:** finite state machine; Delta3D engine; component

由于生活生产成本的不断上升, 真人实际应急救援演练的成本也不断攀升, 不可能满足频繁的模拟救援需求. 为了减少人力物力的消耗, 这就产生了对模拟应急救援系统的需求. 通过模拟演练系统不仅能解决上述问题, 还可以模拟无法动用大量人力实际参与的情形<sup>[1]</sup>. 如辽河流域发生突发污染或者泄露事件等, 都很难进行大规模的模拟演练, 同时还能通过该系统模拟训练救援人员和起到对救援人员指挥的作用, 还能通过该系统中的多部门联合演练功能检验多部门配合的能力和通过推演功能来验证救援方式是否可行. 因此, 构建模拟演练系统势在必行.

应急救援模拟演练系统是基于 Delta3D 游戏引擎设计和实现的. 在模拟演练系统中用户控制最多的就是系统中的角色, 所以角色是整个系统至关重要组成部分. 在模拟演练系统中, 角色之间以及角色和用户的相互通信导致了各个角色状态的不断变化. 特别的在推演过程中由客户生成演练剧本后, 系统将自动执行模拟演练, 在这期间任何的角色都不受客户的控制, 将完全交给系统自行解析更新角色的状态. 这就要求采用一个合理且通用的方式解决角色间相互影响, 相互作用所引起的状态变化的问题. 本文提出了采用 FSM (有限状态机) 技术为核心的 FSM 控制器组

① 基金项目: 国家水体污染控制与治理科技重大专项课题经费资助(2012ZX07505004)

收稿时间: 2012-07-06; 收到修改稿时间: 2012-09-17

件系统, 解决模拟演练系统中角色的状态的转换和控制<sup>[2]</sup>.

## 1 有限状态机原理

### 1.1 基本原理

有限状态机是游戏中人工智能最常用的技术, 其他的状态推理技术基本都需要有限状态机的支持<sup>[3]</sup>. 有限状态机把角色的行为抽象成若干种状态, 通过转换条件, 相互转化. 例如在模拟演练系统中, 当应急人员向危险源移动时, 到达污染点, 转换成治污状态; 当处理完毕, 他会转向空闲状态.

有限状态机是由有限的状态组成的一个机制. 一个“状态”就是一个状况. 考虑一下门的状态. 它的“状态”有“开”和“关”以及“锁”与“未锁”. 对于一个有限状态机, 它应该有一个输入, 这个输入可以影响状态转换. 有限状态机应该有一个简单(或复杂)的状态转换函数, 这个函数可以决定什么状态可以变成当前状态.

如图 1 是一个开关门的状态转换. 圆圈内在为状态, 门在两个状态中切换; E 标识进入状态是的动作; 箭头表示转换条件.

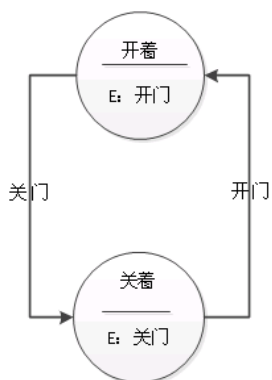


图 1 状态图

上图中, 动作的控制是复杂状态机的最重要部分. 目前动作可分为四类: 进入动作(在进入状态时执行)、转移动作(在特定转换时执行)、输入动作(依赖与当前状态和输入条件)和退出动作(在退出时执行).

除了上图的状态转换图外, FSM 还可以使用状态转换表来表示.

状态机还可分为确定型(DFA)和非确定行(NDFA、GNFA)自动机. 在确定行的状态机中, 每一个状态对每一个可能的输入只有一个精确的转移. 非确定型自动机中, 一个状态对给定的输入可以没有或者多个状

态. 这个在实践中而非理论中更有用.

### 1.2 有限状态机的优势

有限状态有第一个优点就是 FSM 可以明确 NPC (non-player-charater 非玩家角色)的行为, 这样可以为每一类角色设计具体的状态和状态逻辑, 任何没有这方面经验的开发者只需一点基本知识就可以实现. 第二点就是可预测性, 通过一组输入和已知的当前状态, 状态转移就可以预测并且可以简单测试. 由于他们的简单性, FSM's 可以很快的设计、实现和执行.

有限状态机的相关理论已非常成熟, 在游戏中作为状态推理的核心技术已经得到了验证, 并且有很多的例子可以去学习.

像大多数技术, 何时和如何实现 FSM 是主观的和具体问题而定的. 有限状态机适合的领域是很明确的, 就是很容易的通过状态转换表和状态图来表示, 并且具有一套定义良好的状态集和规则集来管理状态转换.

## 2 模拟演练系统FSM设计

### 2.1 状态图的设计

#### 2.1.1 状态集设计

在具体设计模拟演练系统中的有限状态机前, 需要先确定角色的动作和事件, 设计出状态转换图. 由于角色比较多, 此处以环境应急人员为例. 根据模拟演练系统对应急人员的需求分析, 大致可以提取出如下的状态集合:

- ①空闲(应急人员没有被选中)
- ②移动(应急人员移动)
- ③投放净水剂(应急人员向污染河流投放净水剂)
- ④封堵河流(应急人员围堵受污染河流)
- ⑤死亡(应急人员生命值为 0, 角色消失)

#### 2.1.2 事件集设计

通过状态集可以设计出状态转化条件所需的事件, 即前面介绍的动作, 这里将上面的投放净水剂等活动称作技能. 大致有如下事件:

- ①鼠标点击移动的目的地
- ②技能键被触发(由鼠标点击技能图标或键盘触发)
- ③技能释放结束
- ④应急人员生命为零(应急人员处在危险源中)

这里的事件设计的不是很复杂, 主要是作为参考,

而在程序中, 我们需要更加细化这些事件, 完善它的触发逻辑.

### 2.1.3 事件和状态转换图

通过前面设计的事件集和状态集合, 我们就能画出一个简单的环境应急人员状态转换图了, 如下图 2 所示:

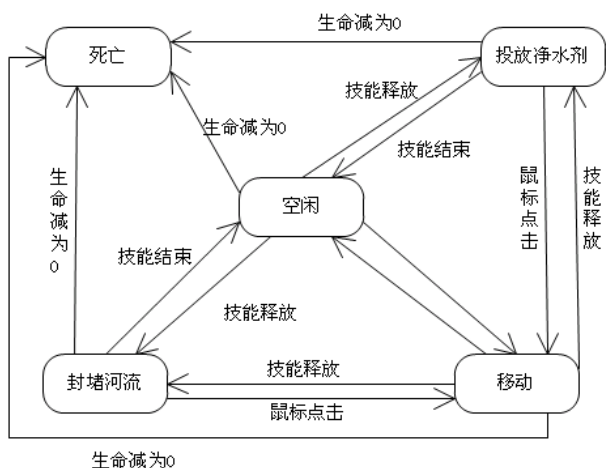


图 2 环境应急人员状态图

初看这个图时会觉得特别的复杂, 但是使用状态机来编码实现起来非常的清晰. 可以根据这个状态图通过程序实现一个带有权值的有向图, 权值就是状态转换需要判断的事件条件的代码, 节点的值下文提到的状态类的代码. 这个图是由 FSM 控制器组件系统动态加载 XML 配置文件实现的.

## 2.2 模拟演练系中 FSM 控制器组件系统的设计

### 2.2.1 Delta3D 核心架构简介

由于本文中的模拟演练系统是基于 Delta3D 游戏引擎设计开发的, 所以离不开 Delta3D 的架构<sup>[4]</sup>. 本文主要研究的是有限状态在模拟演练系统中的应用, 所以只要关注 Delta3D 中最主要的消息、角色和组件, 因为这三者跟 FSM 控制器组件系统设计是相关的, 不用关心 Delta3D 中其他部分的东西.

游戏世界中的角色对象其反应有两个基本的方式: 通过积极的观察世界(轮询), 或者通过等待消息(事件驱动). 在模拟演练系统中会有 10 个部门, 每个部门都有自己不止一个的角色, 面对这么多的对象唯一合理的方案就是采用事件驱动技术.

由于目标角色为了响应事件驱动的行为, 就需要设计一个强大的通信系统, 而 Delta3D 恰恰为我们提供了这么一个强大的通信系统, 即游戏管理器. 我们

通过消息可以在角色、组件以及游戏管理器之间进行通信. 消息可以用于任何角事物之间通信, 比如环境应急人员角色可以有选择的接收他关心的消息包括组件和其他角色的消息, 而对于组件来说它强制接收游戏管理器发送的所有消息.

所以根据 Delta3D 引擎中游戏管理器的特点, 我们把有限状态机中的事件设计成一个复杂的事件消息对象而不单单是事件码或消息码, 以适应不同对象间相互作用的参数传递问题, 也方便在网络上一体化发送<sup>[5]</sup>. 这个复杂的消息体一般会携带角色的位置信息, 状态信息等.

### 2.2.2 FSM 控制器组件的设计

在研究了一些有限状态机的实现方式<sup>[6]</sup>并基于 Delta3D 游戏引擎和模拟演练系统的特点, 本文把有限状态机设计成一个 Delta3D 的组件(及下图 3 中的 FSM 控制器组件), 用于接收和处理角色发送的消息并将结果发送给游戏管理器, 并由游戏管理器发送给角色. FSM 控制器组件与游戏管理器在 Delta3D 中的关系如下图 3:

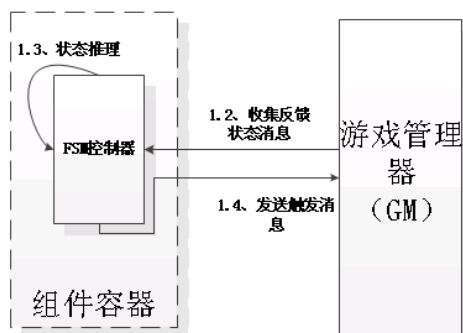


图 3 FSM 控制器组件与游戏管理器关系图

当我们的角色发送事件消息的时候, FSM 控制器组件会接收这个消息, 通过推理后, 以消息的形式把新状态发送给游戏管理器, 由游戏管理器通过消息角色映射表, 发送给需要该消息的角色, 使角色具有新的行为. 例如当用户鼠标点击环境应急人员移动时, 环境应急人员把初始位置状态信息及角色 ID 等封装成消息对象发送给 FSM 控制器组件, 通过 FSM 控制器组件推理计算后把新位置发送给该角色, 环境应急人员收到消息更新位置信息实现移动.

FSM 控制器组件的设计中, 不会使用 switch-case 模式去实现状态之间的转换, 虽然这种方式简单, 但是也有设计缺陷, 不能很好实现面向对象, 状态改变

也得修改源码,不能很好的动态添加新状态到状态机中.所以我们会把 FSM 控制器组件设计成三个部分:第一部分是 FSM State 类部分,主要用于封装具体的状态实现;第二部分是 FSM Machine 类,就是主要的状态机推理类;第三部分是 FSM Control 类,主要用于消息接收、发送和处理,管理 FSM Machine 类和 FSM State 类.三者的关系如下图 4:

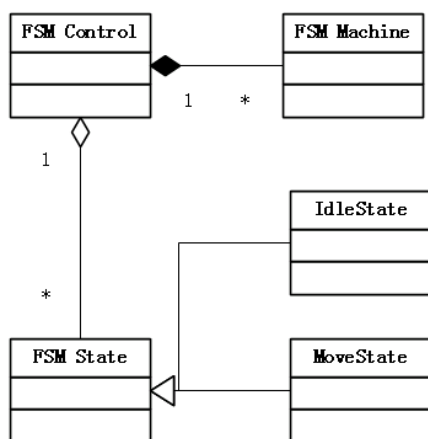


图 4 FSM 控制器组件逻辑架构图

下面将分别介绍这个部分的设计.

首先 FSM State 类的设计,我们不是将它限制在每一个角色中,而是将每一个状态设计成类,每一个状态会有自己的算法去处理当前状态需要做的事,例如当选中一个环境应急人员并点击一个位置让它移动时,FSM Machine 根据状态图推理出相应的状态代码,FSM Control 类根据代码调用上图中的 MoveState 状态,这个状态实例会接收角色的初始位置和目标位置,通过状态内部封装的算法,得出该角色移动的位置是否可达,以及移动路径等信息并把结果封装成事件消息对象发送给角色.同时,通过将一类状态设计成一个状态类,我们可以不用为每一个角色的状态都设计一个类,例如模拟演练系统中的环境应急人员,消防员等都有移动状态,我们只需要实现一个 FSM State 类就可以了,达到了复用的目的也可是使我们的状态类的数量在合理范围内.

由于一个推理机只能真对一类对象,所以在系统中,会为每一种角色设计一个具体的推理机类.FSM Machine 类的调用是发生在 FSM Control 类中的,如果一类角色处于初始状态,则会为该类角色创建一个特定的有限状态机类的实例.它定义了一个状态对象列

表:模拟演练系统中角色的 ID(由 FSM Control 提供)该角色的当前状态的代码、当前输入事件的代码,如果某个角色的状态达到了终止状态即死亡状态,FSM Machine 类把它从列表中移除,还定义了一组有关状态机对状态图的操作.这样的设计的好处是在系统中会有多个环境应急人员,而系统只会创建一个 FSM Machine 类的实例.FSM Machine 实例中用于推理的状态转换图的生成是通过读取 XML 文件实现的,例如下图是根据图 2 的状态图编写的环境应急人员的状态图配置文件的一个节点,如图 5 所示:

```

<TransitionList Name="HuanJingYingJiRenYuan">
  <Transition Type="state" name="BlockRiver">
    <Property Name="event" Value="MouseClicked" />
    <Transition Type="NEXT_NODE">
      <Property Name="Move" Value="MouseClicked" />
      <Property Name="Idle" Value="SkillStop" />
      <Property Name="Dead" Value="0" />
    </Transition>
  </Transition>
</TransitionList>
  
```

图 5 封堵河流节点的 XML 文件示例

通过这种设计,就可以对状态图进行添加和修改而不用修改程序.有了这个配置文件,FSM Machine 类就会为环境应急人员生成 FSM 状态图并进行状态推理了.这样添加和删除状态转换逻辑只需修改配置文件而不用修改源码.FSM Machine 在每次推理后会返回目标状态代码,并将结果发送给 FSM Control 类.

FSM Control 类的是整个 FSM 控制器组件的核心部分,也是被模拟演练系统调用的类,它需要以组件的形式在系统中注册.FSM Control 类会接收游戏管理器发送过来的消息,进行解析,根据角色和 FSM Machine 类的映射关系,创建和维护 FSM Machine 类的实例.FSM Control 类还负责管理 FSM State 类,如果演练系统添加了新的 FSM State 类只需在 FSM Control 类中注册就可以添加到系统中了.同时,FSM Control 类在接收到角色消息时,首先会对消息体进行解析,判断是哪类角色,之后会到角色、FSM Machine 映射表中寻找该类角色的对应的 FSM Machine 实例.角色和 FSM Machine 实例的映射是通过在 FSM Control 类中注册后实现的,如果有新的角色和新的状态机只要注册到 FSM 控制器组件,无需修改程序.最后 FSM Control 类把状态信息打包发送给游戏管理器,由游戏管理器发送给相应的角色处理.

(下转第 124 页)

示.

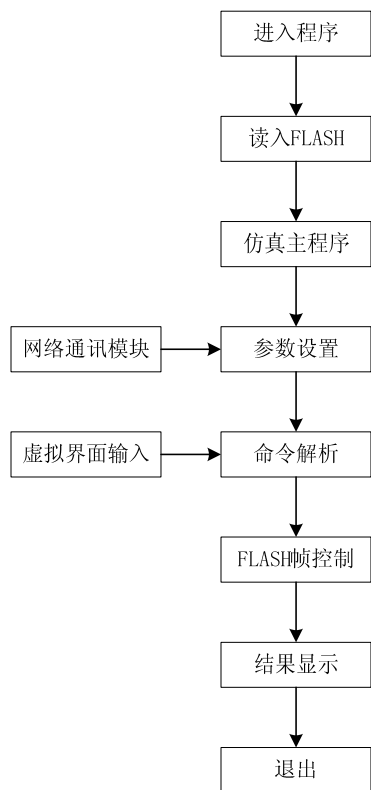


图 4 软件流程图

### 4 结语

潜艇雷弹发射装置模拟操作软件的开发与设计完全遵照面向对象分析、面向对象编程, 综合利用 VC、FLASH 和 3DSMAX 软件开发模拟操作软件, 获得了良好的效果, 能够显著降低训练费用, 提高训练效率, 使操作人员迅速了解发射系统的工作流程, 掌握操作方法. 该方法可用于同类模拟训练系统或其他系统的操作控制台的软件开发与设计.

### 参考文献

- 汪致远. 海军武器装备. 北京: 原子能出版社, 2003.
- 周冬华. 模拟训练系统的操作控制台软件开发与设计. 计算机科学与技术, 2005, (26): 11.
- 郭克新. Visual C++ 代码参考技巧大全. 北京: 电子工业出版社, 2008.
- 刘锐宁, 宋坤. Visual C++ 开发典型模块大全. 北京: 人民邮电出版社, 2009.
- 余家劲. 某型航空控制弹药仿真控制台的设计. 计算机仿真, 2005, (34)3.
- 刘航, 王春水, 王积忠. 基于视景仿真技术的某装备虚拟操作训练系统. 指挥控制与仿真, 2007, 29(2).

(上接第 103 页)

### 3 结语

有限状态机在基于 Delta3D 引擎的模拟演练系统中的应用是采用组件的方式设计和实现的, 使得状态机更加结构化和利于维护, 系统中角色的状态变化可控, 条理更清晰. FSM Control 对 FSM Machine 实例的统一维护解决了对对象间相互作用时产生的同步问题, 同时对于状态的设计更具有复用性, 例如移动状态可以适用于所有的角色. 同时我们为每一个状态赋予了更强的功能, 将一些复杂的算法和逻辑封装在状态中而不是角色类中, 提高了系统的灵活性减少了角色的复杂度. FSM 控制器组件系统可以将所有过去的状态信息与引起每一个状态变迁的原因一起存储到一个文件中, 采用这种方式设计方便了程序的调试和对有限

状态机设计的验证.

### 参考文献

- 孙成江, 刘林. 应急救援模拟演练系统设计与实现初探. 石油工业计算机应用, 2010, (3): 3-6.
- Brownlee J. Finite State Machine (FSM). 2002.
- Deloura M. 游戏编程精粹 1. 北京: 人民邮电出版社, 2004.
- Rosenblum L, Macedonia M. Project in VR. The Delta3D Open Source Game Engine, 2005.
- 白晓梅, 王茹, 赵云兵. 基于 Delta3D 的气象仿真框架设计与实现. 现代电子技术, 2012.
- 邹循进, 叶云青, 陈孝威. 第三届和谐人机环境联合学术会议 (HHME2007) 论文集. 2007.