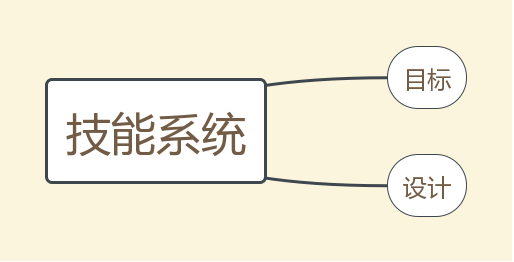
这篇文章总结一下最近对技能系统的学习。本文主要分为两个部分，一，技能系统的设计目标；二，具体设计。



# 1概念

## 什么是技能系统

技能系统是一个封闭的系统，它的内部结构和实现对外部不可见，它接受技能请求，并给出正确的输出，如动画，特效等。



## 技能系统的复杂性

1 同一个角色有多个技能。

1. 场景中有多个角色在释放技能
2. 怪物也有技能。
3. 状态多变。攻击状态，空闲状态。受击状态。击飞。眩晕。
4. 技能单体，范围。
5. 带锁定，不带锁定。
6. 技能还可以被打断
7. 与服务器交互
8. 技能的施放分多个阶段

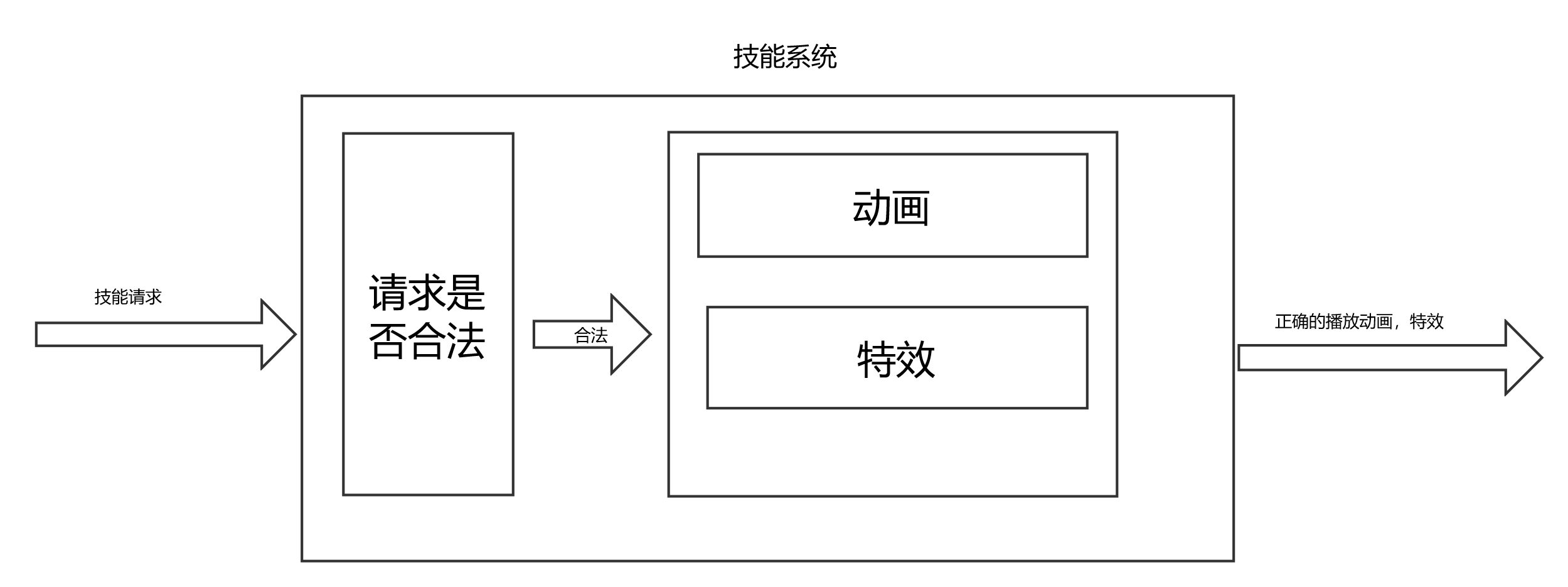
这样就造成了技能系统的规则复杂，效果多样。因此我们必须好好分析一下对于技能系统而言，它的输入输出应该有怎样的结构。对于技能系统的技能请求必须有技能ID，施放者，接受者。而对于技能系统的输出具有多样性，它可能有一个角色动画，有一个特效，产生一个buff，产生一个弹道，产生一个法术场。

## 1.3 好的技能系统应该解决什么问题

一个好的技能系统应该**具有尽可能少的入口**，**正确的处理技能的施放，产生正确的效果，并且结构清晰，模块之间低耦合**。

# 2 设计

要正确的处理技能的施放，首先必须对技能请求进行合法性判断。所以一个技能系统系统内部应该分为两个阶段。一，技能合法性判断，二，技能施放。



然而即使技能请求是合法的，要正确的施放技能也具有相当大的复杂性。技能的请求多种多样，这样就导致了技能逻辑中出现大量的if else 语句。这种语句出现多了容易导致代码的可读性变差，后续的修改也会变的困难。而且一旦技能规则变得复杂会导致各个 if else 之间出现交叉或者冲突等等问题。这个问题的解决方法便是状态机，在设计模式中也称作状态模式。

在设计具体的状态机之前我们先总结一下技能系统的技能请求会有哪些类别。

1. 运动请求,Idle，move，run，jump，fly等
2. 技能请求
3. Buff请求
4. 法术场请求
5. 弹道