DashFire技能系统——类天龙设计方案

草稿版

1. **核心思想：**

游戏里，每个新英雄的推出，依靠的都是

1. 独特的技能——（策划&程序）
2. 拉风的造型——（策划&美术）
3. 略有深度的故事背景。（每个参与制作的人员）

所以我们的技能系统至少要有以下两点要求：

1. 能稳定效率的正常运转
2. 随意添加一个与之前任何技能都可能毫无关系的新技能，还能保持1的状态

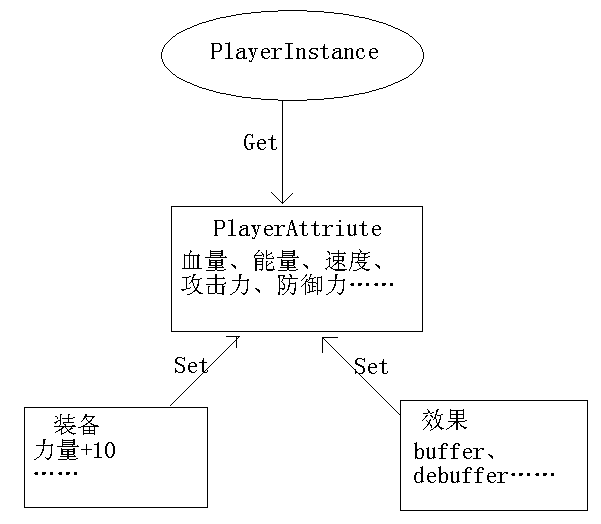
※想要做出一款精品游戏，工作量和时间是不可能节省的，每推出一个新英雄都需要策划、美术、程序做大量的工作，一劳永逸的设计在DashFire里可能不会很容易做到

1. **与技能系统相关的一些基础模块**

首先服务器现在分为哪些模块，每个模块之间什么关系，怎么通信我还没有完全想清楚，所以这里先列出一些与“技能”相关的3个基础模块：Attribute、Impact、Skilllogic。这些模块的分类和天龙很像，这也是为什么我称之为“类天龙方案”：

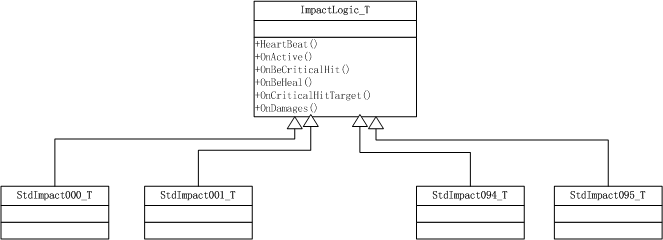
1. PlayerAttriute 人物属性模块

所有的人物属性，都保存在这里，这里只负责存储数据，不负责逻辑运算。它与其他相关模块的交互如下图所示：



1. 装备系统会影响玩家数据的变化；玩家身上的多个效果也会影响他的数据变化。
2. 游戏场景里可能一次会保存10个PlayerInstance，只有PlayInstance才可以随时Get到它自己的Attriute
3. 战斗的过程就是【玩家在客户端噼啪敲键盘】—>（省略一些细节）—>【服务器上每个PlayerInstance的数据在不断改变】—>【反馈给客户端】
4. Impact 效果模块
5. 所有的效果都派生自同一个基类，这个基类没有数据成员、只有成员函数。

效果类结构图如下（盗天龙的图）：



1. 每个派生类表示一个效果逻辑，用来实现这个效果是如何如何对这个玩家的Attribute产生这样那样的影响。是一个可以发挥策划和程序想象力的存在。
2. 肯定会有一个ImpactManager来负责管理Impact的添加、移除以及每一个Tick的循环，暂时觉得每一个PlayerInstance有一个自己的ImpactManager比较好。
3. SkillLogic 技能逻辑模块
4. 所有的技能都派生自同一个基类，这个基类没有数据成员，只有成员函数。
5. 派生类通过重载不同的虚函数来体现自己的特性，比如“瞬发技能”，“引导技能”，“蓄力技能”，“被动技能”等不同的子类。
6. SkillLogic模块，主要负责技能【是否能够成功释放】，【释放！】，【释放中。。。】，【释放结束】，【被打断】，【对\*\*产生了\*\*效果】等逻辑处理。
7. 要存在一种脚本技能逻辑机制——ScriptSkillLogic，用来处理一些特殊的（比如Boss或者某RMB玩家相同的技能比较特殊的处理等问题）。
8. 每个PlayerObj下面应该有5-10个技能槽，用来存放他的这几个技能——SkillLogic，可以通过它的Instance来触发该SkillLogic的执行。
9. 每个技能的冷却时间也是这个PlayerInstance的一个Attriute。
10. 释放一个技能，服务器端的流程图大致如下:

