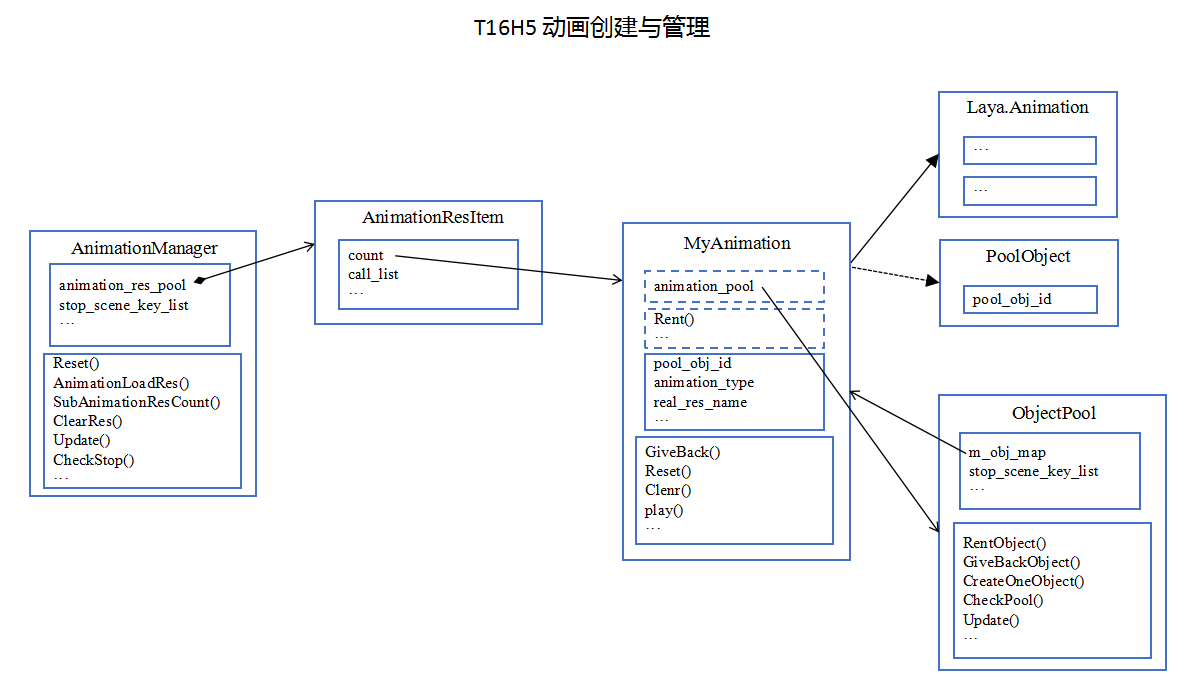
**T16H5动画创建管理模块**

# 1、Laya图集帧动画创建接口

# 2、动画池

## 2.1、动画池结构

首先为了减少游戏过程中对象频繁创建与销毁带来的设备性能的消耗，而通过对象池对对象的创建与销毁的管理来达到提高游戏运行效率的目的。在T16H5项目中，动画池模块主要由AnimationManager、MyAnimation、ObjectPool组成，相关类图如下所示：



其中ObjectPool为主要抽取出来的动画池相关属性，因为除动画对象外还有其它如UI、地图等对象的创建与销毁也需要对象池的管理，其主要属性有确定池保存对象的类型，保存对象的列表，对象列表中空闲对象的标记列表，现有使用和空闲对象统计，以及防止池过大而设置达到某个阈值时自动清理阈值，同时相对的设置池在低于某个阈值时不再清理防止池对象全部清完而再使用时还得创建，而方法主要有从池中取出对象，创建对象，归还对象，池状态更新，如在update中实现池的对象的摊帧创建防止对象在某帧上大量创建造成游戏卡顿，以及通过update实现对象池的自动化清理。

在T16H5中MyAnimation继承Laya.Animation类并扩展了一些属性和方法，主要有ObjectPool对象及动画实例的名字、类型和对象像在池中ID等属性。而因为该对象主要起着保存具体动画对象属性的作用放于池中进行管理，如动画的位置，大小，中心点，所以感觉把animation\_pool和相关取存方法防御manager中合理些。

而AnimationManager类主要为管理动画资源的创建和管理。因为当一个动画资源加载后，可能游戏的很多地方都要用到该资源，而并不是游戏中某个点不用动画后就把涉及的动画资源卸载，为此增加了AnimationResItem类和在AnimationManager中增加游戏资源池来管理某个游戏资源的引用，以便在不用时卸载游戏资源优化内存。其中涉及游戏管理的方法又有资源创建，update更新管理，清理资源，暂停池中某些特定或全部动画播放更新操作。

## 2.2、对象池管理

当通过对象池缓存暂时不用的对象来提高设备CPU计算性能时，如果有大量的闲置而不清理时，会对内存造成浪费。因此需要及时动态的对池中对象进行清理，最大化利用内存。而动画对象池的管理分为动画对象MyAnimation的管理和动画资源AnimationResItem的管理，主要通过ObjectPool和AnimationManager中的update进行。

# 3、动画创建与销毁过程

## 3.1、动画创建

## 3.2、动画销毁