一、特性

1. 资源自动管理。在一定时间没有引用的资源，会被销毁。
2. 资源引用计数：加载引用和实例化引用。
3. 资源最近访问时间。
4. 资源销毁策略：手动，自动（没有引用，且一段时间内不被使用）
5. 协程并发限制

二、协程数据结构

1. 概述

并发的协程数量可以控制。

1. 数据结构
2. Task:执行或者等待的任务。

Id:任务id

void OnLoaded():加载成功后的回调

{

}

1. void OnLoaded():加载成功后的回调
2. 实现细节

三、资源管理数据结构

1. ResLoadTask：资源加载任务，加载完毕后回调Resource
2. ResUseCallback：资源使用的回调函数类型
3. UseCallBack

Callbackffunc

Resource res;

1. LoadRef：对资源的加载引用，在加载后，会根据实例化策略转化为实例化引用

Callback

InsStrategy

1. InsRef：对资源的实例化引用

Callback

1. Resource: 资源数据结构，对应一个已经加载或者正在加载中的资源。

{

Float lastestVivitTime;//最近访问时间

ResStatus;//资源状态，加载中or加载完毕。

ResLoadTask loadTask;//加载任务

List<LoadRef> loadRefList;//加载请求

List<InsRef> insRefList;//实例化请求

}

1. Map<string,int> pathIDDict;资源路径和资源id的词典。
2. Map<int,Resouce> idResourceDict; 资源id和资源的数据的词典。
3. List<Use

核心逻辑

Int GetResource:

string path,

enumCortinueType corroutineType,

enumClearStrategy clearStrategy

enumInstantaiteStrategy insStrategy

CallBackFunc callBack

加载中或者已经加载完毕

If pathIDDict.contians(path)

nResId = pathIDDict.contians(path)

Resource res= idResourceDict [nResId];// 获取资源

res.UpdateVisitTime();

If(res.IsLoading)

{加载中

Res.uselist.Add( new UseCallBack( CallBaCallBackFunc, insStrategy) );

}

else

{

已经加载完毕

Instantiate(Res,CallBackFunc, insStrategy);

}

return nResId；

else

Resoucce res = NewSource(path, clearStrategy, insStrategy );

res. loadTask = LoadingCorroutine.singleton.GetLoadSession(corroutineType).Load(LoadTask);

res.uselist.Add( new UseCallBack( CallBaCallBackFunc, insStrategy) );

return res.Id;

endif

NewSource

String path

enumCortinueType corroutineType,

enumClearStrategy clearStrategy

enumInstantaiteStrategy insStrategy

CallBackFunc callBack

RescourdId++;

Resouce res = new Resouce(ResoucedId, path, corroutineType, clearStrategy, insStrategy, callback)

pathIDDict[path] = RescourdId;

idResourceDict [RescourdId] = res;

return res;

实例化相关

Instantiate

Resource res, CallBackFunc callback, enumInstantaiteStrategy insStrategy

If( insStartegy == Immediate)

Callback(res)

If( insStartegy == Delayed)

AddInstanceQueue();