使用（PocketRPG文档的ReadMe中有提及）

1.Calling StartTrail(float timeToTweenTo, floatfadeInTime) andFadeOut(float fadeTime)will fade in and fade out the trail respectively.

调用StartTrail和FadeOut函数去实现渐显示和渐消失的效果。时间参数以1秒为一个单位。

2.Calling SetTime(float trailTime, floattimeToTweenTo, float tweenSpeed)can change the length of the trailinstantly, giving you a little more control.

调用SetTime函数去改变拖尾长度

3.The WeaponTrail can be built by callingItterate(float itterateTime) andUpdateTrail(floatcurrentTime, float deltaTime). These functions are called byAnimationController, however if you don't want to use AnimationController youcan call these yourself.

调用Itterate和Update函数去更新拖尾效果，目前还不明白这两个函数的参数怎样添好，幸运的是，PocketRpg的开发者做了AnimationController脚本去代替我们调用这两个函数使拖尾效果更好。

AnimationController：

http://img.my.csdn.net/uploads/201301/16/1358323991_6890.png

控制动作，使拖尾效果更好

参数：

Gather Delta Time Automatic：

使用

1.调用AddTrail函数去添加受影响的WeapontTrail对象。

2.调用PlayAnimation函数去切换动作。

------------------------------------------------------------------------------------------------------------------------------------------------------------

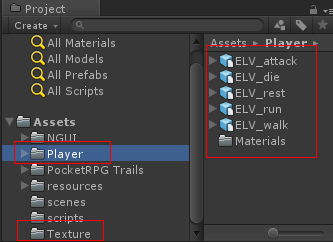
应用：

模型：天堂的剑士（网上下载的）

动作：攻击和休息

模型包：http://download.csdn.net/detail/xv\_ly15/5001911

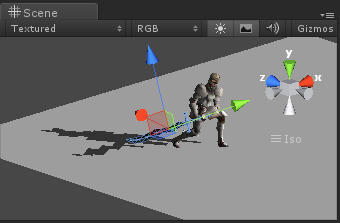
导入模型包后，能在Unity编辑器下看到Player和Texture两个文件夹。



在player目录下能看到各个动作包。

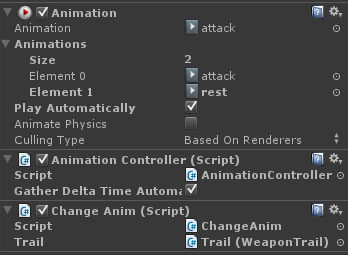
用Attack和Rest做武器拖尾效果演示。

将Attack拖到场景中



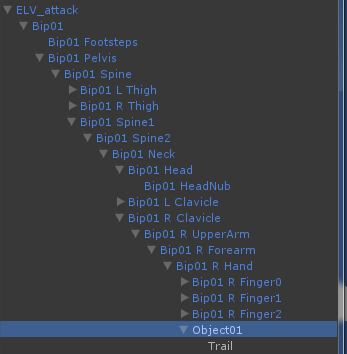
添加AnimationController脚本

添加AnimationController脚本，对象已带有Attack动作，再通过Rest对象添加Rest动作。

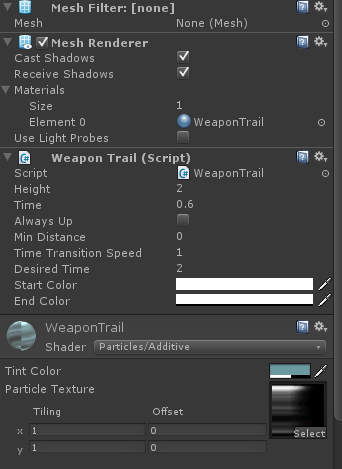


添加WeaponTrail脚本

找到需要添加拖尾效果的GameObject，添加Trail对象用于实现拖尾效果



Trail添加必须的Mesh Filter，Mesh Render，拖尾材质还是使用PocketRPG自带的刀光材质。



使AnimationController和WeaponTrail生效

新建一个脚本，假设命名为ChangeAnim用于改变动作并控制拖尾效果。与AnimationController同一等级赋到GameObject上。在上面添加AnimationContoller的图片中能看到。

在ChangeAnim下生命如下四个成员变量

    //带有WeaponTrail的对象  
    public WeaponTrail trail;  
    //攻击动作  
    private AnimationState animAtk;  
    //休息动作  
    private AnimationState animRest;  
    //动作控制脚本  
    private AnimationController animCtl;

声明后将Trail对象拖到该脚本使用的Trail中，同样在上面添加AnimationController的图片中能看到。

在Start函数中添加如下代码

        //获取动作控制脚本  
        animCtl = gameObject.GetComponent<AnimationController>();  
  
  
        //获得攻击和休息动作  
        animAtk = animation["attack"];  
        animRest = animation["rest"];  
  
  
        //添加受动作控制脚本的拖尾对象  
        animCtl.AddTrail(trail);

在Update函数添加如下代码，响应A，B按钮控制动作

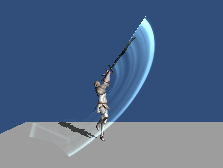
        if (Input.GetKeyDown(KeyCode.A))  
        {  
            //播放攻击动画并打开拖尾效果  
            animCtl.PlayAnimation(animAtk);  
            trail.StartTrail(0.5f, 0.4f);  
              
        }  
        else if (Input.GetKeyDown(KeyCode.B))  
        {

   //播放休息动画  
            animCtl.PlayAnimation(animRest);

            //渐消失  
            trail.FadeOut(0.5f);     
        }

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

效果



http://blog.csdn.net/xv\_ly15/article/details/8509781