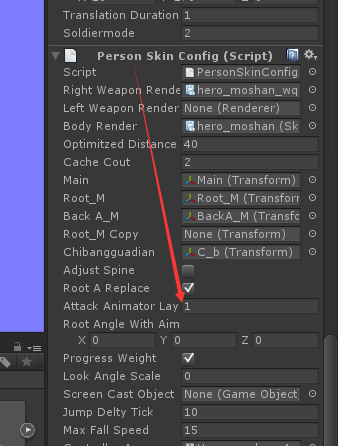
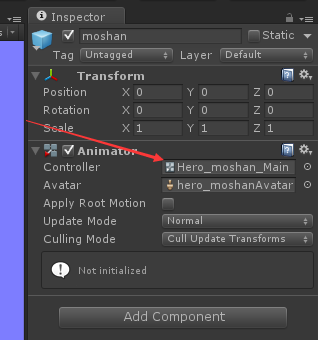
第三版动作配置说明

1、预制体

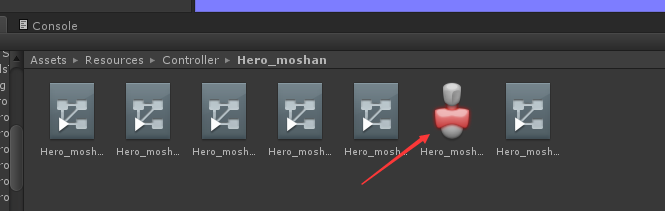


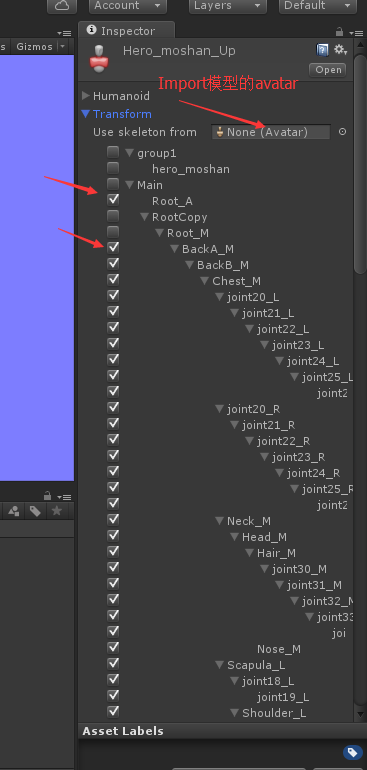
AttackAniamtorLayer设置成攻击动作所在的第一层的索引，如果索引为0，则为旧版ABC状态机融合机制



设置成配置好的新版animatorController

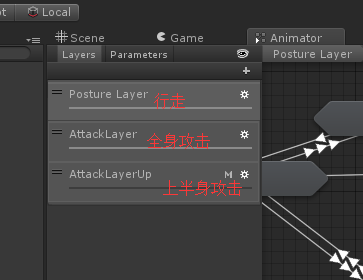
1. 新建一个上半身的模板 AvatorMask



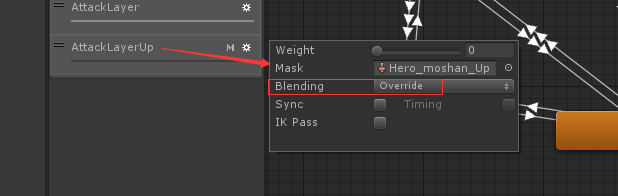


勾选上半身需要的节点

1. 改animatotController



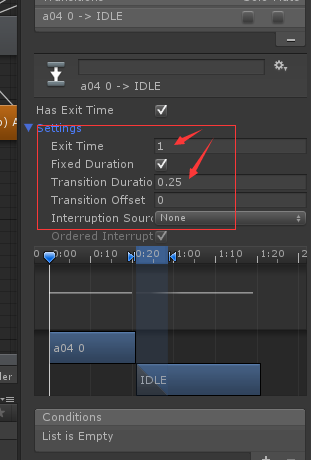
全身攻击和上本身攻击，动作直接复制



上半身加mask

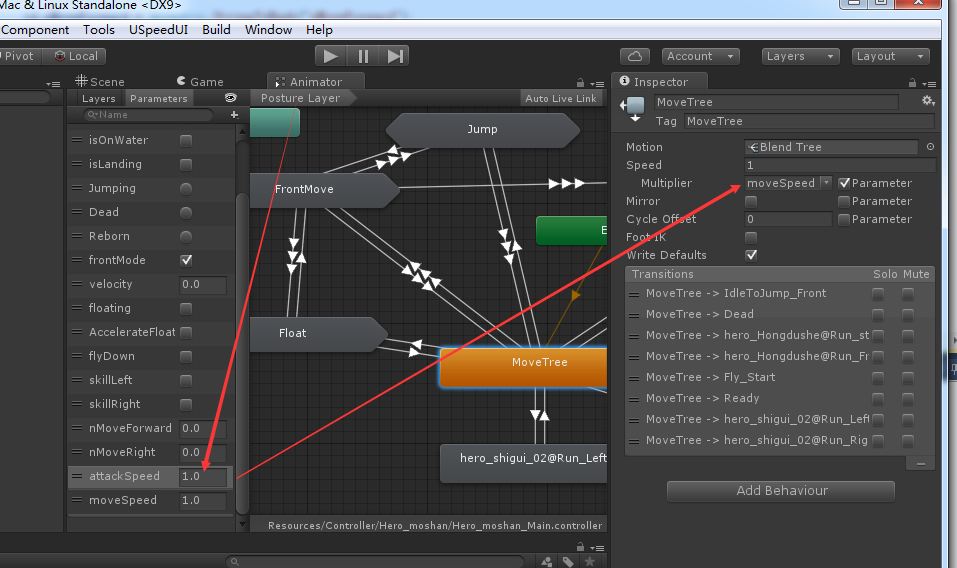
如果索罗斯等行走动作有两层，则AttackAniamtorLayer=2

1. 微调攻击层



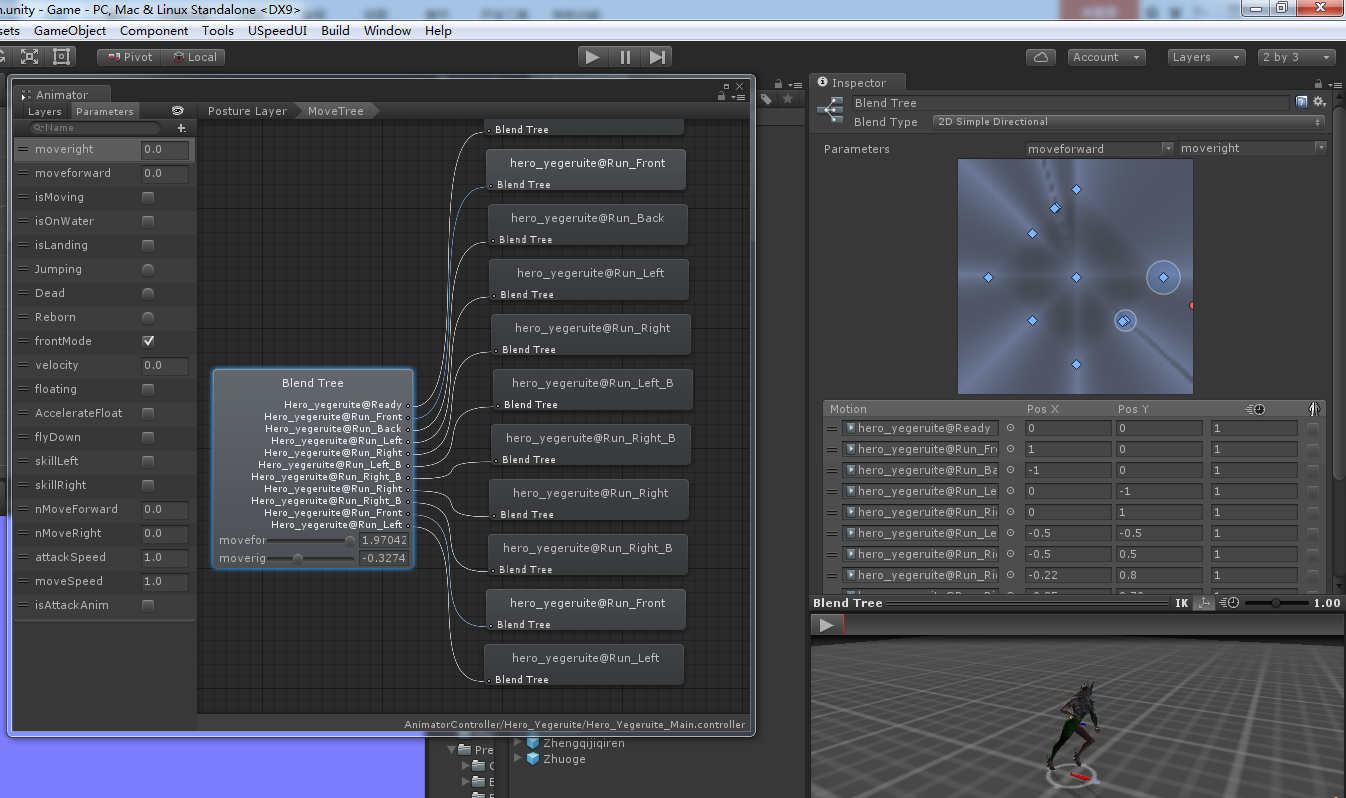
边走边打动作，和IDLE之间的过渡，从1开始，过渡时间可以调整

1. 增加速度控制

  
增加控制参数float类型的attackSpeed和moveSpeed，默认配置1.0

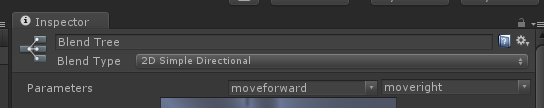
为移动动作的Speed Multiplier配上参数moveSpeed；为攻击动作配上attackSpeed

1. 移动动作融合配置说明

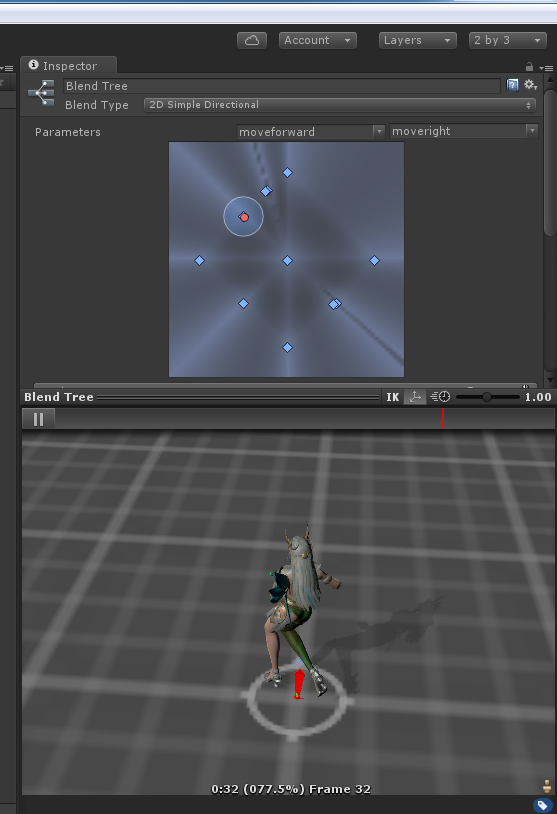
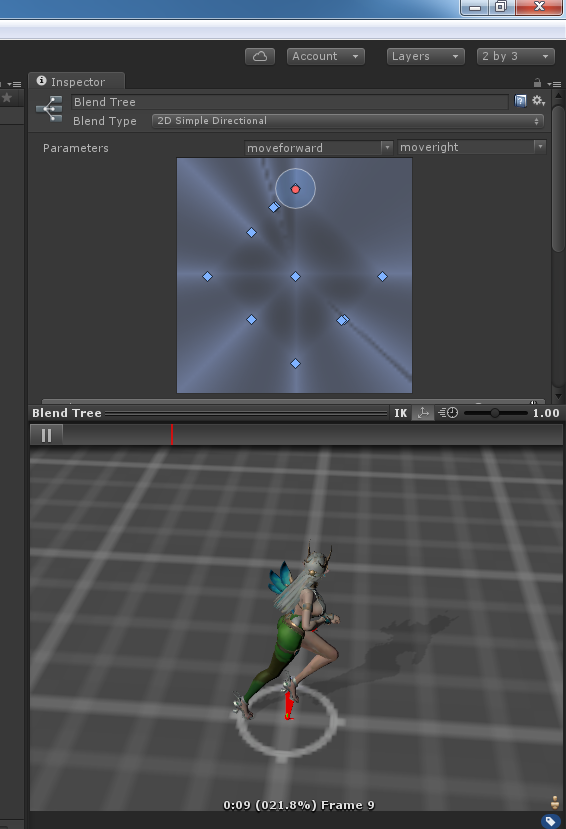


移动的时候根据不同的移动方向合成移动动作，采用的是Unity自带的BlendTree，

上图是弓箭手的配置例子。

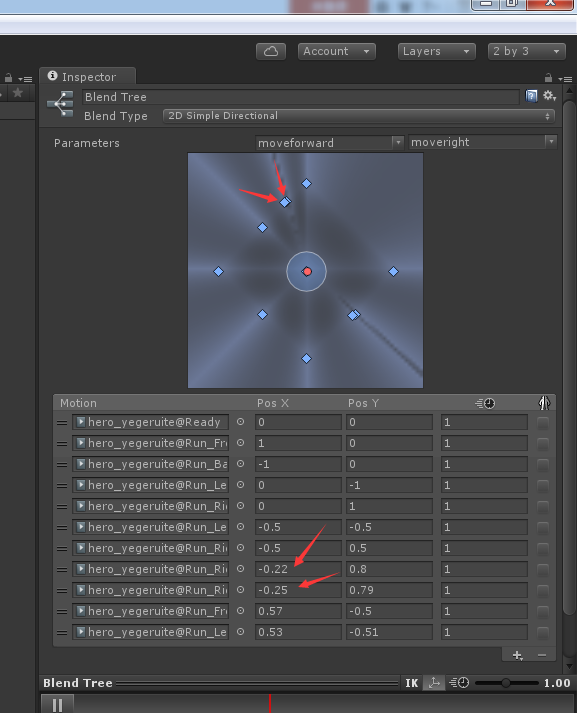


可以看到动作融合的类型选的是2D Simple Directional，所以动作是根据实际参数（红色点）在XY坐标轴上的角度取最近的几个动作插值而成的。



但是简单的插值融合会容易造成扭曲。例如左图弓箭手向右跑步的动作，是腿部向右做前进动作；而右图弓手右后跑步的动作，是腿部向左做后退从左；因此当前进方向介于向右与向右后时，两个动作直接插值会融合出一个并腿滑步的动作。

解决方式是取一个中间角度，插入两个动作，中间角度偏右一点点的取向右动作，中间角度偏后一点点的去向后动作，最终配置如下图：



另外，还需注意，两个插入点需要尽量近，才能尽量较少出现错误插值的角度范围。但是又不能靠的太近，因为太近的时候Unity会认为是同一个角度，无法区分。当角度差异过小时Untiy会报警，如下图

