在unity里面，情况也是如此，不过再复杂一点展开。  
1、骨架就是transform。不像maya和理论那样，是一种实体的单独的存在。在unity里，骨架就是transform的层次关系。一个骨头就是一个transform。这个人体的骨架（Transform）组织结构一般是abdome（spine,LeftUpLeg,RightUpLeg），abdome是根（有时用hip），向上是spine（neck，LeftUpArm,RightUpArm）。这个在哪里都差不多。关键是root，就是abdome。在unity里，引用骨架都是靠它。  
2、Mesh。因为Unity里的Mesh有绑骨和未绑骨两种（在哪里都差不多）。在导入资源的时候，unity会为绑骨的Mesh自动生成一个SkinnedMeshRenderer，为未绑骨的生成一个MeshFilter和一个MeshRenderer。不精细的人物，头盔、手、武器（甚至靴子）等都不用绑骨，像[骑马与砍杀](https://www.baidu.com/s?wd=%E9%AA%91%E9%A9%AC%E4%B8%8E%E7%A0%8D%E6%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YknWndPjnLn1cvmyc4ujTd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0LPWm1rHf3PWf1nHm1n1nsr0)，只需要把armor绑骨就好。  
不绑骨情况：MeshFilter负责从文件中取出真实的Mesh（这里用的是Script里的类来表示，具体可以查reference）。MeshRender负责显示。这种想要固定在人体上，[其实很简单](https://www.baidu.com/s?wd=%E5%85%B6%E5%AE%9E%E5%BE%88%E7%AE%80%E5%8D%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YknWndPjnLn1cvmyc4ujTd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0LPWm1rHf3PWf1nHm1n1nsr0)，把这个GameObject的tansform的parent设置为需要绑定的骨头的transform就好。（比如武器，transform.parent=item\_l，item\_l是左手一个绑定武器骨头）所以，缺点也很明显，一个mesh只能绑定到一个骨头，也就是这个mesh不会变形。（这个可以作为绑骨和不绑骨的依据，看Mesh是否需要变形，所以头盔，武器都不需要绑骨（仅在unity中成立））  
绑骨情况：SkinnedMeshRenderer有俩个重要成员，一个是Mesh，一个是Bones。当然，Mesh里有每一个顶点的绑骨数据，但这里存的是序号。就是第几个骨头，并没有指明是哪个骨头，但是如果给一个根骨头就能推测出所有的骨头了。这里的Bones把所有的骨头都列出来了，至于顺序，有点扯淡，没啥规律（调试结果）。问题出来了，那SkinnedMeshRenderer里的骨头这么没顺序，Mesh里的骨头序号怎么对应过来呢？好吧，仔细想想，其实没那么复杂。因为SkinnedMeshRenderer里包含Mesh和Bones两个成员，由于都是一起生成的，那么就一个坑埋一个人，正好就对应起来了。之所以要这么个Bones列表，是因为要把它画成transform树吧。  
这样就再回答一个问题，如何换装备？原SkinnedMeshRender对应的骨骼已经存在，不可能销毁它（还有其他部分要用），所以就修改SkinnedMeshRenderer本身就好。要改的有两个值（加上材质三个）：Mesh和Bones。其实，原Bones根本不用变，只需要把Mesh变了就好（除非你的绑骨不是一帮人做的）。详细代码看看Unity有个官方项目。另外，我觉得吧，连SkinnedMeshRender都不用操纵，直接重新生成就好。看文章最后的代码。  
动画部分最有爱。其实呢，layer这个玩意，用来分组比较合适。优先级这个特点也是可以用的，不过寄太大信心。一个layer可以设置同样的叠加方式（累加，混合），控制方式（只控制上半身）。一般不同动画都放在不同层，相同功能动画在同一层，比如向左砍人和向右砍人。另外，我们动画一般都是自己搞到的一个一个片段，如果你不是动画师，还是不要用啥累加和混合了。用一些独立的动画，让他控制身体的某些部位就好。  
所以，综合看，人物控制，在unity 里，换装备，换武器都是可行的。不知道咋了，网上就是搜不到如何换装的，搞到我自己研究了好久。。。  
另外，最近的新发现，升级到4后，有个mecanima模块，用来做动画重定位的，这个叫帅啊。我的以下代码没办法处理动画，因为动画之前无法重定位。但是可爱的unity的新功能来了，给每个armor和每个动画都创建一个对应的avatar，然后就直接可以用一套动画驱动所有人了（虽然表面上我们显示的是一个人，但是在内部，实现这种换装方法用的是好几个独立的人体骨骼，当然，这种方法要求每个独立的骨骼都是相同的）。  
  
附上换装代码。  
#pragma strict  
public var O\_body\_r:GameObject;  
public var O\_hand\_L\_r:GameObject;  
public var O\_hand\_R\_r:GameObject;  
public var O\_helmet\_r:GameObject;  
public var O\_calf\_L\_r:GameObject;  
public var O\_calf\_R\_r:GameObject;  
public var O\_weapon\_R\_r:GameObject;  
  
public var body1\_r:GameObject;  
  
private var mBody:GameObject;  
  
private var mHand\_L:GameObject;  
private var mHand\_L\_P:Transform;  
  
private var mHand\_R:GameObject;  
private var mHand\_R\_P:Transform;  
  
private var mHelmet:GameObject;  
private var mHelmet\_P:Transform;  
  
private var mCalf\_L:GameObject;  
private var mCalf\_L\_P:Transform;  
  
private var mCalf\_R:GameObject;  
private var mCalf\_R\_P:Transform;  
  
private var mWeapon\_R:GameObject;  
private var mWeapon\_R\_P:Transform;  
  
function Start () {  
  
}  
  
private var ifIs:boolean=true;  
function Update () {  
  
if(Input.GetButtonDown("Fire1")){  
Generate(ifIs);  
ifIs=!ifIs;  
}  
  
}  
  
function Generate(num:boolean){  
if(mBody){  
Destroy(mBody);  
}  
if(num==true){  
mBody=Instantiate(O\_body\_r,transform.position,Quaternion.identity);  
}else{  
mBody=Instantiate(body1\_r,transform.position,Quaternion.identity);  
}  
  
mHand\_L\_P=mBody.transform.Find("abdomen/spine/thorax/shoulder.L/upperarm.L/forearm.L/hand.L");  
mHand\_L=Instantiate(O\_hand\_L\_r,mHand\_L\_P.position,Quaternion.identity);  
mHand\_L.transform.parent=mHand\_L\_P;  
mHand\_L.transform.localRotation=Quaternion.Euler(0,270,90);  
  
mHand\_R\_P=mBody.transform.Find("abdomen/spine/thorax/shoulder.R/upperarm.R/forearm.R/hand.R");  
mHand\_R=Instantiate(O\_hand\_R\_r,mHand\_R\_P.position,Quaternion.identity);  
mHand\_R.transform.parent=mHand\_R\_P;  
mHand\_R.transform.localRotation=Quaternion.Euler(0,270,90);  
  
mWeapon\_R\_P=mBody.transform.Find("abdomen/spine/thorax/shoulder.R/upperarm.R/forearm.R/hand.R/item.R");  
mWeapon\_R=Instantiate(O\_weapon\_R\_r,mWeapon\_R\_P.position,Quaternion.identity);  
mWeapon\_R.transform.parent=mWeapon\_R\_P;  
mWeapon\_R.transform.localRotation=Quaternion.Euler(0,270,90);  
  
mCalf\_L\_P=mBody.transform.Find("abdomen/thigh.L/calf.L");  
mCalf\_L=Instantiate(O\_calf\_L\_r,mCalf\_L\_P.position,Quaternion.identity);  
mCalf\_L.transform.parent=mCalf\_L\_P;  
mCalf\_L.transform.localRotation=Quaternion.Euler(0,270,90);  
  
mCalf\_R\_P=mBody.transform.Find("abdomen/thigh.R/calf.R");  
mCalf\_R=Instantiate(O\_calf\_R\_r,mCalf\_R\_P.position,Quaternion.identity);  
mCalf\_R.transform.parent=mCalf\_R\_P;  
mCalf\_R.transform.localRotation=Quaternion.Euler(0,270,90);  
  
mHelmet\_P=mBody.transform.Find("abdomen/spine/thorax/head");  
mHelmet=Instantiate(O\_helmet\_r,mHelmet\_P.position,Quaternion.identity);  
mHelmet.transform.parent=mHelmet\_P;  
mHelmet.transform.localRotation=Quaternion.Euler(0,270,90);  
  
}