**【完美国际】玩家变身**

## 策划需求

玩家在游戏中可以进行特殊的的变身操作，变身后的玩家外形改变。在不同的变身状态，有不同的动作限制。

## 数据来源

玩家数据结构info\_player\_1中的GP\_STATE\_SHAPE用来表示玩家当前形态。

变身时客户端收到服务器发来的PLAYER\_CHGSHAPE协议，包含当前的GP\_STATE\_SHAPE值。

## 程序结构

形态数据shape为8bit，各个bit含义如下：



其中TYPE取值范围即EC\_Player.h中定义的枚举值：

**PLAYERMODEL\_\***

而Model ID为EC\_Resource.h中定义的模型ID：

**RES\_MOD\_\***

非0的TYPE表示玩家处于变身状态，由于只有2bit，所以变身状态共有3种：

**TYPE1表示职业变身**

即“妖精，妖兽，刺客，巫师”等职业，变身为“狐狸，老虎和人鱼”的状态。这种状态下玩家基本和正常状态相似，可进行各种动作。但“相依相偎”的动作在职业变身状态下各有不同，客户端根据模型进行了单独处理。

这种类型的Model ID的值如果为1，则客户端会根据玩家当前职业和性别来选择合适的模型。目前服务器恒定Model ID为1。也可添加更多的符合动作需求的模型来产生更丰富的职业变身效果。

**TYPE2表示技能变身**

当前“丛林遗迹”中的金钱蛙变身属于此类。这种状态下玩家动作限制较多，基本上只能进行移动，一段跳跃和普通攻击。

Model ID来自于策划者技能中填写的值。注意模型必须有4种普通攻击动作（可以相同），否则在普攻的时候会出现僵直。

**TYPE3暂时未使用**

（细节待定）

## 注意事项

**Model ID的选择**

在不同的变身状态下，玩家可进行的动作是不同的。所以Model ID对应的实际模型必须有合适的动作，否则就会出现错误。目前可以用在职业变身类型TYPE1中的有：

**RES\_MOD\_ORC\_TIGER = 22,**

**RES\_MOD\_ORC\_FOX = 23,**

**RES\_MOD\_SHADOW\_FISH\_M = 24,**

**RES\_MOD\_SHADOW\_FISH\_F = 25,**

可以用于技能变身TYPE2可用的模型ID有：

**RES\_MOD\_MONEYFROG = 26,**

具体可以参考客户端代码。

**模型的异步加载**

玩家数据初始化的时候，只会加载正常状态的模型。如果一开始就处于变身状态，相应的变身模型也会被加载。之后，随着玩家进行各种操作，至多会加载“不超过变身类型总数”的模型。目前只有三种变身，所以至多会同时加载3个模型。这样的好处是，即使需要支持更多的变身模型，也不会造成大量的模型预加载。

当某玩家变身时，对应的模型会被异步加载。加载完成之前，玩家的当前模型不会改变，也就是说视觉上会有短暂的延迟。

当已变身的玩家进入当前玩家视野的时候，对应的模型会被异步加载，加载完成之前玩家模型不显示。

被加载过的变身模型在下次触发变身的时候，会被立即显示，异步加载不会发生。例如第一次妖狐变会触发异步加载，第二次妖狐变就会立即显示了。

同一类型的变身模型如果发生改变，仍然会触发异步加载。例如第一次技能变身为金钱蛙，第二次变身为兔子，第三次再变身为金钱蛙，这三次都会触发异步加载。但连续两次变身金钱蛙则不会触发异步加载。