

Writing Sample

I started to create my own game called Light in October. Before I started to make this game, I've conceived a lot of content in my mind and referred to a lot of games.

In this game, light is a very important element, which runs through the whole game, just like the soul in Dark Souls series. Players can gather light orb to recover their health and light. Players can also use a certain proportion of light to launch high damage attacks. In my setting, the player as the container of light, the maximum light energy stored on his body is equal to the maximum health value, because each high damage attack consumes a fixed proportion of light, so when the player's maximum health value is larger, the more light he needs to launch a high damage attack, and on the contrary, the less. So you may wonder how to reduce player's maximum health? Here I refer to the riven system in Warframe. Every time player defeats an enemy, the enemy has a chance (50%) to drop a riven. The attributes of each riven are completely random. They may contain positive or negative attributes or both, including the number of attributes. The details are shown in the table 1 below.

Number of Attributes	Number of Positive Attributes	Number of Negative Attributes	Ration (Positive)	Negative (Negative)
1	1	0	1.000	-
1	0	1	-	1.000
2	1	1	0.942	0.755
2	2	0	0.755	-

Table 1 Riven attributes

In a nutshell, this set of attribute system is built in the hope that new players can use high fault tolerance to complete the process, and for skilled players, the challenges brought by higher damage and lower fault tolerance will be more interesting.

The construction of this system took me a lot of time and I completed a lot of challenges. First of all, I created the data structure of riven, and listed all four possible situations in the blueprint of riven, and implemented them in random order. There are also some problems and bugs. For example, when there is an entry related to life value in a drive, when it is positive (i.e., increase the maximum health value), the player's current health value should not increase, because the player can recover by repeatedly equipping and taking off the drive, but when it is negative (i.e., decrease the maximum health value), the player's current health value should not exceed the maximum life value to prevent overflow.

In order to cooperate with this set of attribute system, the two actions of characters (light attack and heavy attack) should be completely separated. Here I refer to Bloodborne and the well-known Monster Hunter. In the normal state, the player's attack using straight sword will not consume any light, and the action is faster and the rigidity is smaller. After consuming a certain amount of light and using deformed attacks, the player will use claymore as a weapon, and the action is as fast as using straight sword, but the rigidity is larger. At this time, the attack will consume a certain amount of light each time, but it can deal great damage. Deformed attacks can parry the enemy's attack and

return the consumed light at the moment of use. When using Claymore, the protagonist can also deform straight sword without consumption.

The design of this set of action system is to highlight the deformed attack. Players can use straight sword mode to consume enemies' health and then capture the right time to change to claymore mode and use light to deal great damage.

However, it is very difficult to fulfill all the requirements of this action system, especially when considering the deformed attack, and determining of the deformed attack's parry is also very difficult to solve. In the end, I had to give up making the deformed attack.