Expectation management of From Software: a key to the open world genre.

Yoko Taro speaks in his 2018 GDC talk, 'Freedom is the moment that previous framework suddenly expand'; In contrast, many open world developers hold a concept 'Large map, Various items, Many quests.' Yoko's concepts enlighten me, and I expand it to find out an efficient way to build an open world. I will use my method to analyze Sekiro in the essay.

For me, Open is giving choices which exceed the players expectation. The key to open world is to keep players expectation under control. There are three steps to utilize the concept in games. The first step is to build expectation. The second step is to satisfy the expectation. The last step is to exceed the expectation.

1. How to build players expectation?

The first step is to build an **accurate recognition of game boundary,** to let them know what you can do and what you can't in the virtual world. These is often related to environment metrics in producing procedure. In Dark Souls, players cannot jump, and From Software make it a metric by ensuring the minimum height of obstacle is higher than the highest height of a long dash. Therefore, the player won't wonder whether he can jump over a knee-high wall. An opposite example is the Witcher. In the Witcher, Geralt can jump, but this builds a false expectation that you can jump over the fence to get to your destination quicker. Geralt fell to death from as short as two floors which contradicts with players recognition toward jumping. Players will be depressed when they found the character doesn't reach their expectation.

2. How to satisfy players expectation?

To satisfy the expectation, we should first know what expectation is. Players expect to experience **anything they can imagine within the boundary,** and imagination is built by gameplay ruleset. To give a clear definition, I would like to use the concept of dimension and spaces in a mathematic way. Suppose a gameplay can be expressed by rule dimension A and B. Suppose A= {A1, A2, A3}and B= {B1, B2}. Then the whole playable space of gameplay is A*B= {A1B1, A1B2, A2B1, A2B2, A3B1, A3B2}. A*B is the expectation because players naturally combine the element in rulesets and imagine the combination in is valid in virtual world.

Knowing this, our goal as a game developer is to maximize your actual experience space until it reaches the ideal playable space. **The larger space you implement, the opener player would feel.** Besides, **the more dimension (new rules) you add, the less open the game is** because the ideal playable space expand while the experience field doesn't. I would like to use an example to illustrate how my model work in Sekiro. If we evaluate its basic gameplay, we will find that it was composite of two basic dimensions. The first dimension is ruleset of Movement and the second is ruleset of Offensive Behavior. And Movement= {Move, dash, jump, hook, swim} Offensive Behavior = {attack, defend}

IdealGameplaySpace	Move	Dash	jump	hook	swim
Attack	Normal attack	Dash attack	Jump Attack	Hook Attack	Swim Attack
Defend	Normal defend	Dash Defend	Air Defend	Air Defend	Swim Defend

You can see that From Software have implement all the space in the ideal gameplay space. This leads to a result that most of the action player expected can be experienced in game world. Imagine a player learn that Sekiro can hook, and if he notices that Sekiro can attack, he will naturally try whether he can attack while hooking. Then he found his hypothesis is right because Sekiro indeed can hook attack. As a result, the player would find that developers satisfied his expectation and finally would praise the game for open experience. In a nutshell, to satisfy the player expectation, you should **implement the gameplay space constructed by your core ruleset as much as possible**.

3. How to exceed the expectation?

The third step is to exceed the player's expectation. This step is clear in the model in step 2. In Sekiro, From Software expand the ruleset Movement with an additional Movement after player defeats the boss Corrupted Monk. In short, the expanded Ruleset Movement= Movement + Additional Movement. And Addition Movement={dive}. Of course, they add dive attack and defend as well. In this way, Sekiro break and exceed the cognition that formed in the previous environment. Before the diving skill is learned, players would never imagine that Sekiro can explore under water. As a Result, the developers exceed the expectation, and players find that the game is more open than they formerly thought.

From Sekiro, we could primarily verify my method developed from Yoko's concept. With such methods supporting, I am confident with open experience in indie game because open experience has little to do with high cost, while you need to carefully manage the expectation instead.