can't we do the exact opposite?

like random image?

User Avatar

Haha

mint function goes in an order right? Like 1, 2,3 and etc

so if we randomise that people will get confused like why 5 is coming before 3 or 2 is coming after 7 and so on

so instead of this we can pick a random image from ifps and assign in to token ids as they come in order - if this is doable

if not we need to think about a creative way to randomise it

Pls find out if there’s a way to pick random images from ifps and assign to token ids

i saw it on some nft projects where the nft image is revealed randomly after mint

User Avatar

I will read more about it in the morning, we can figure it out

Just this question; in the current logic, the tokenids (1,2,3..100) are minted sequentially or like random as in any number comes at any place?

Just one thing - and this is not a change, this is how I wanted it from the beginning. Minting must be creating tokens with ids in sequential order; like #1, #2 etc as it happens.

I have a simple solution for this - we can mint token ids sequentially - I will mix the images (and metadata accordingly) in the folder to give it a random feel - so you won't really have to worry about randomising it if that's the difficulty you are having right now.

I will upload a new IFPS folder with all 100 images in a random order and will give you the URI; you can simply mint them 1, 2, 3, with correct token ids - that way no mixing up or randomising is needed from your end.

In the current logic, tokenids are minted randomly.  
I think there is no difference between current logic and your new idea.

Both of these two cases, displaying image randomly is the same to players.  
You think, what will be the difference to users.  
In current logic, random images are displayed to users such as #2, #8, #1, #35, etc.

In your logic, also random images are displayed to users like the same as above.

Since no different effect to user, why do you change all metafiles needlessly?

Of course, no problem in my side, but you may be boring for this work.