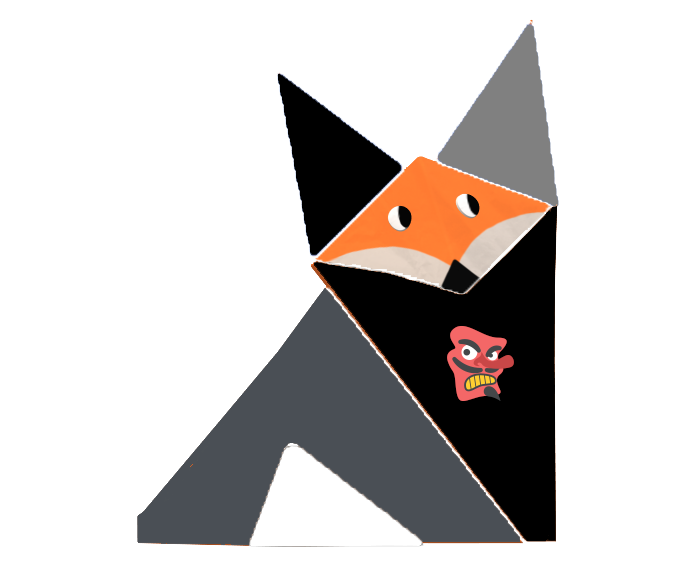
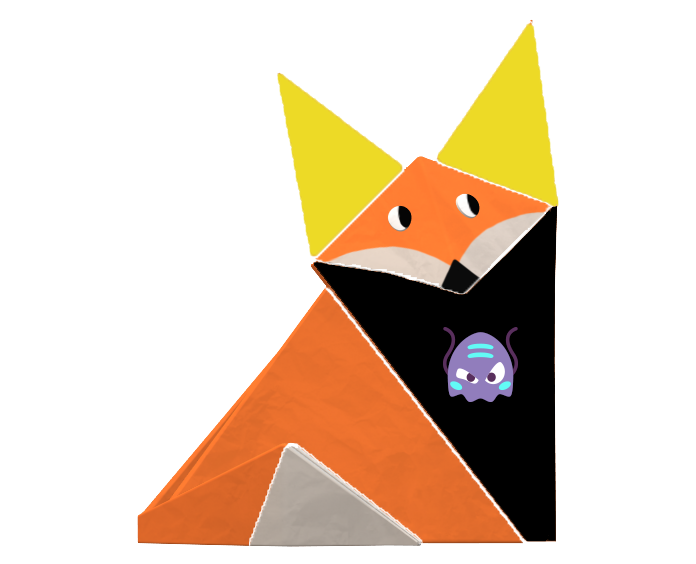
**OrigamiFox NFT**

**@OrigamiFoxNFT**

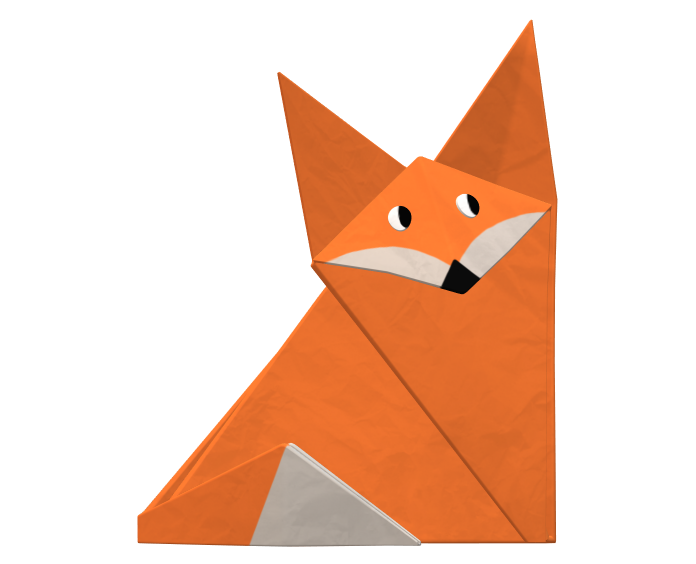
**10,000 randomly generated NFT foxes**

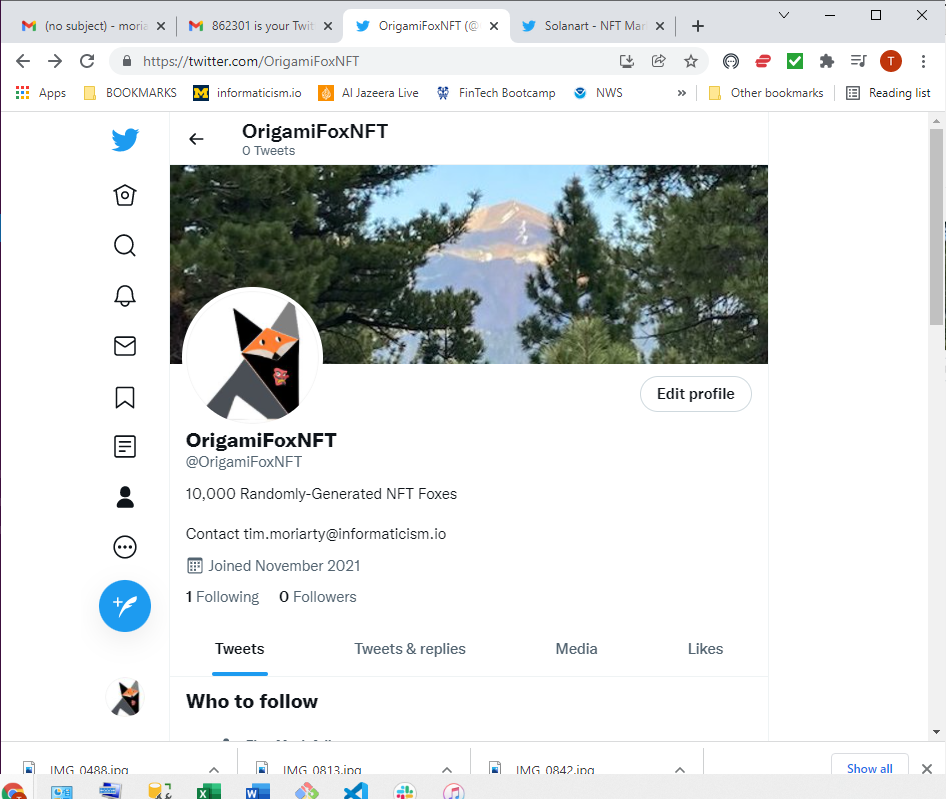
**Brought to you by**

**AQUIBA benarroch, DARIUSH ruch-kamgar, DAVID dannenberg, & TIM moriarty**



**EXECUTIVE SUMMARY**

* Our objective has been to tokenize an asset using Ethereum Request for Comments (ERC) standards.
* Our decision was to create a non-fungible token (NFT) using ERC-721 standards, and to build a decentralized app to facilitate the purchase of a unique piece of art.
* ****We decided upon an existing .png image from Microsoft’s Paint 3D library “3D Origami”. Our specific choice was “Origami fox” to use as a template for the NFT.
* Our initial scope was to create a series of 10,000 unique OrigamiFox NFTs. We ended up with creating an initial set of 819 distinct pieces of art.
* Members of the bootcamp cohort and instructional/support team will have the opportunity to mint their own OrigamiFox NFT.

**OrigamiFox NFTs:**

A collection of 819 unique avatars, each with a distinctive

set of traits.

The traits consist of an initial set of four trait categories:

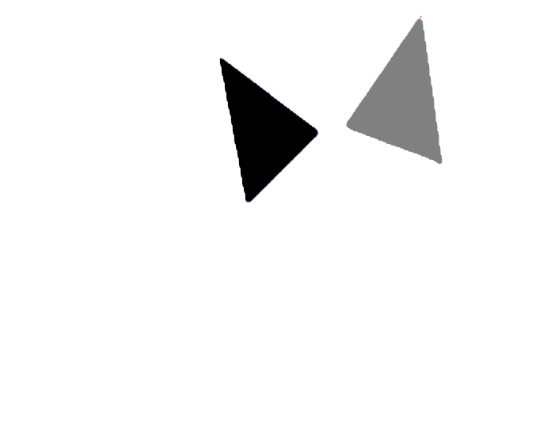
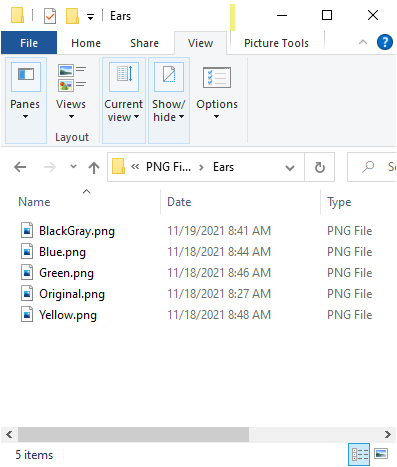
* Ears
* Face / Tail Combo
* Upper Body
* Lower Body

Each trait category contains 5 – 9 distinct combinations of colors and designs, comprising a total of 25 images saved as .png files.

The unique NFTs are created by stacking one .png file from each of the four trait categories. Each of the 25 traits has a “rarity weight” which dictates how often a given trait will appear in an NFT. The following pages show each of the distinctive traits’ .pngs, along with related info for each.

**EAR TRAITS:**

**BlackGray.png**



**TRAIT CATEGORY:**

Ears

**REQUIRED:**

True

**RARITY WEIGHTS:**

20,

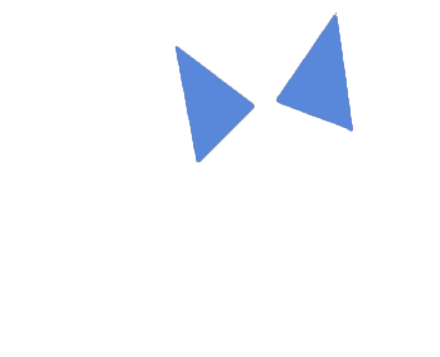
20,

20,

20,

20

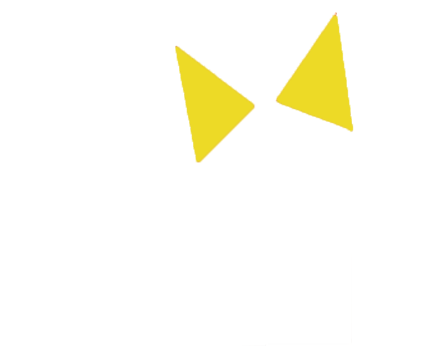
**Blue.png**

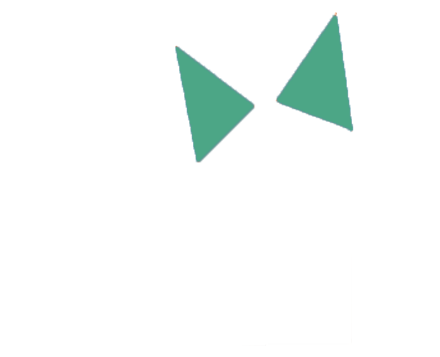


**Yellow.png**

**Original.png**

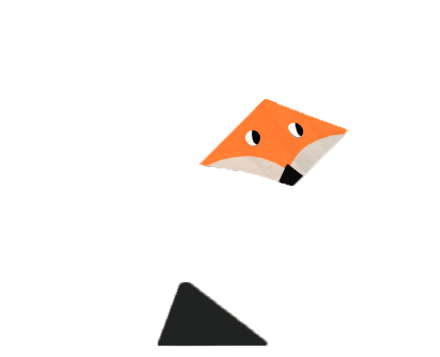
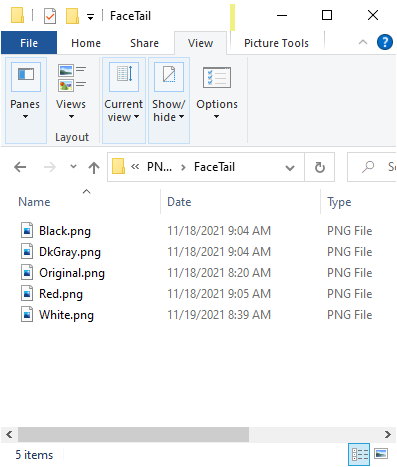
**Green.png**





**FACE/TAIL TRAITS:**

**BlackGray.png**



**TRAIT CATEGORY:**

FaceTail

**REQUIRED:**

True

**RARITY WEIGHTS:**

22,

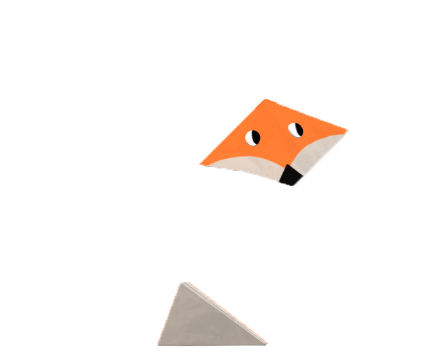
22,

22,

22,

12

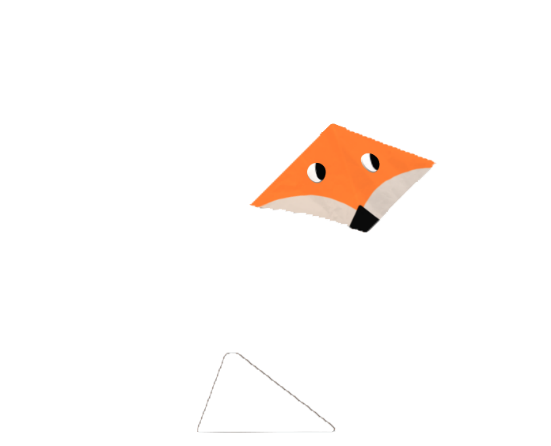
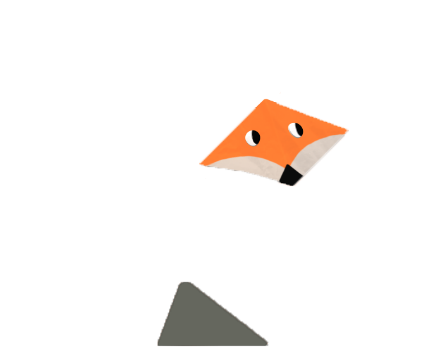
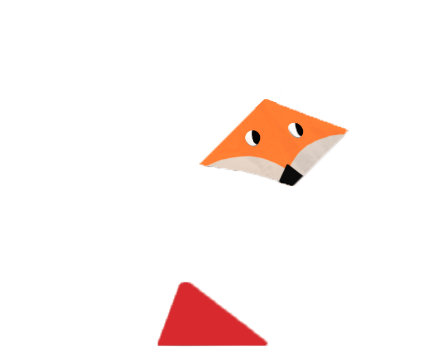
**Original.png**



**DkGray.png**

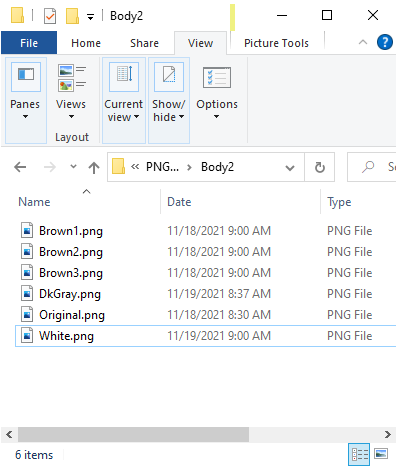
**Red.png**

**White.png**



**LOWER BODY TRAITS:**

**Brown1.png**



**TRAIT CATEGORY:**

Body2

**REQUIRED:**

True

**RARITY WEIGHTS:**

18,

18,

18,

18,

18,

10



**Brown2.png**

**Brown3.png**

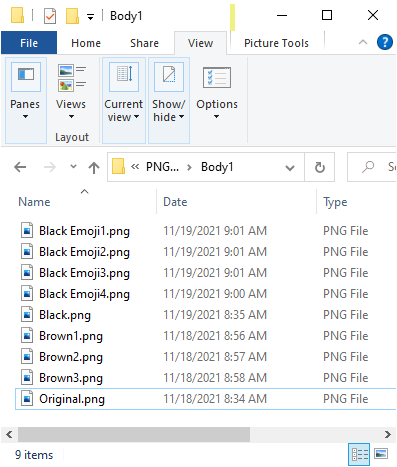
**White.png**

**Original.png**

**DkGray.png**

**UPPER BODY TRAITS (standard):**

**Brown1.png**



**TRAIT CATEGORY:**

Body1

**REQUIRED:**

True

**RARITY WEIGHTS:**

5,

5,

5,

5,

16,

16,

16,

16,

16

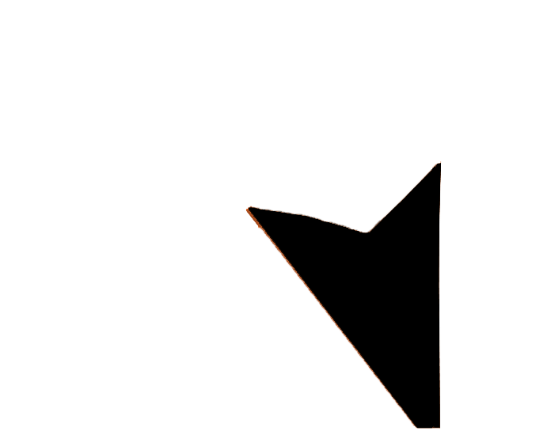
**Brown2.png**



**Original.png**

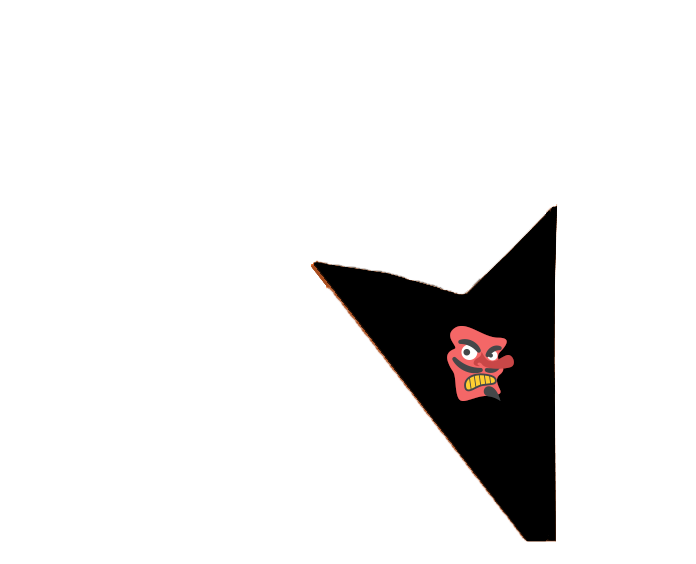
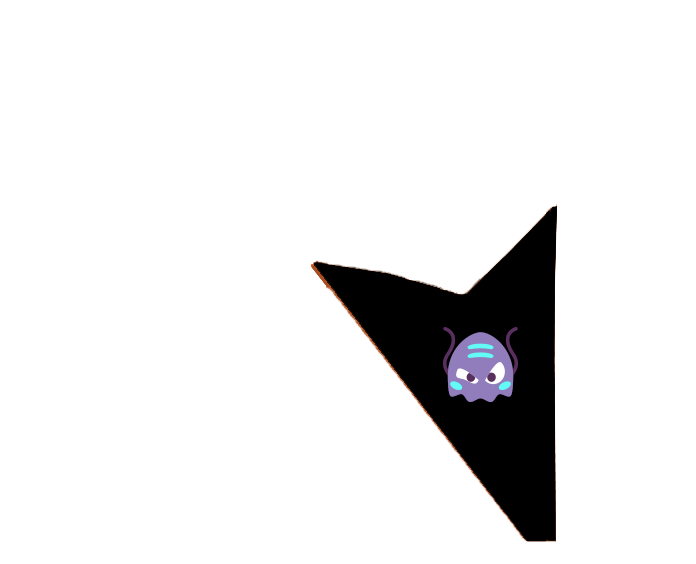
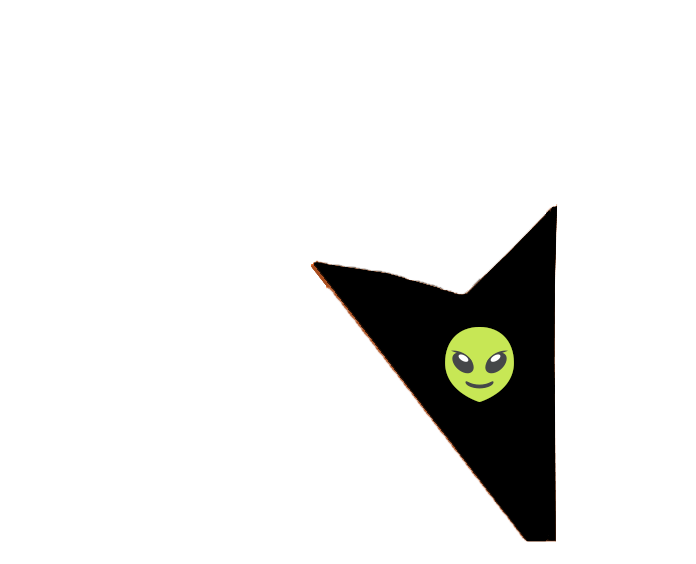
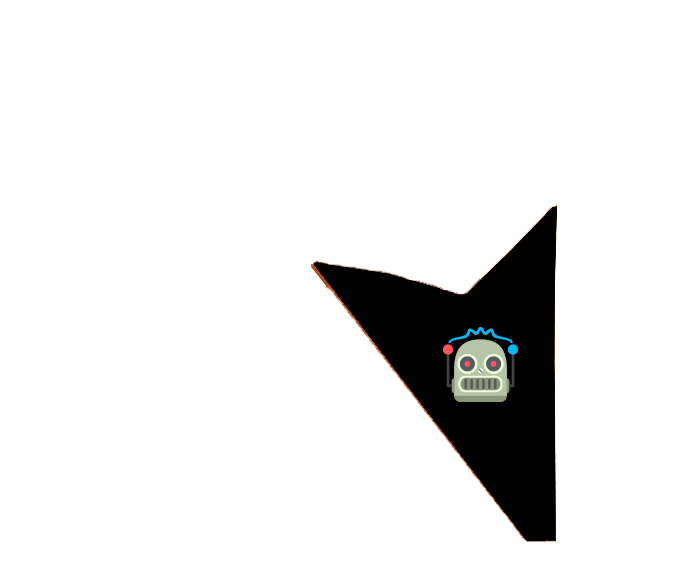
**Black.png**

**Brown3.png**





**UPPER BODY TRAITS (with emoji glyphs):**



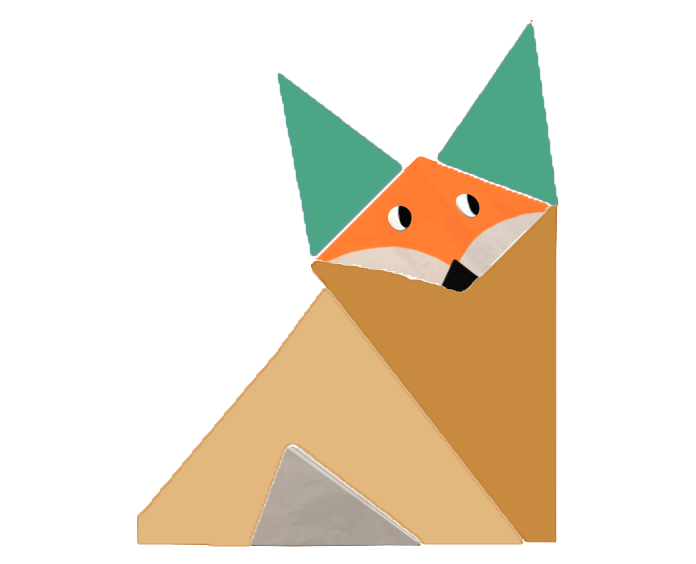
**BlackEmoji4.png**

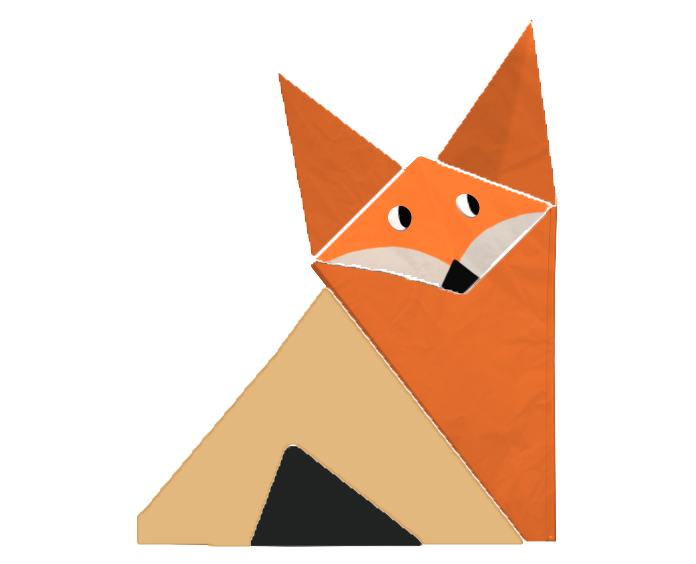
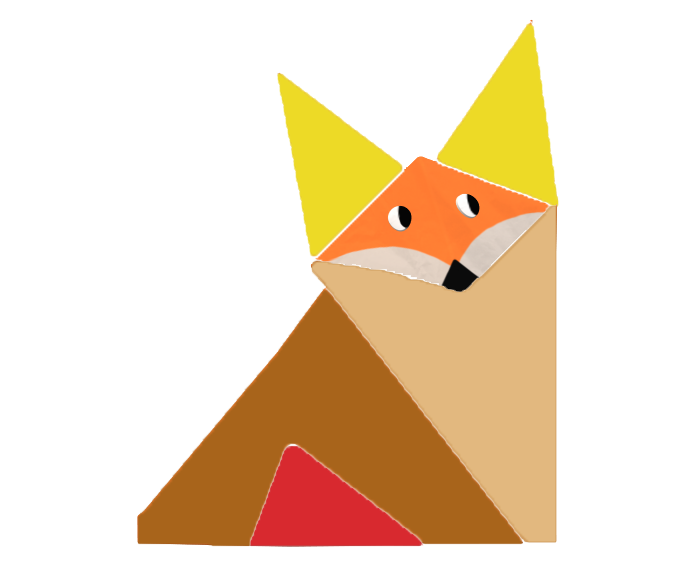
**BlackEmoji3.png**

**BlackEmoji2.png**

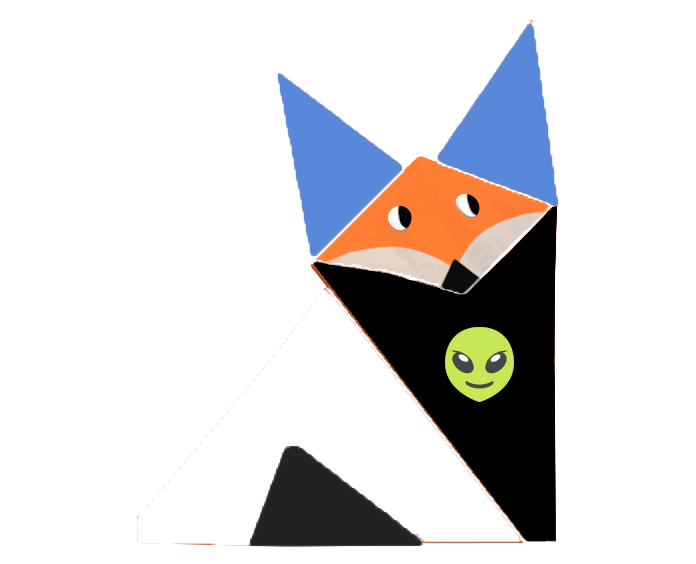
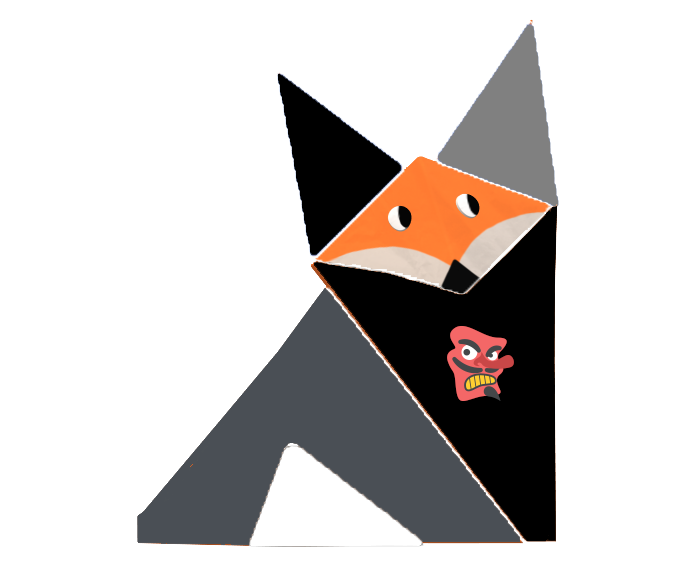
**BlackEmoji1.png**

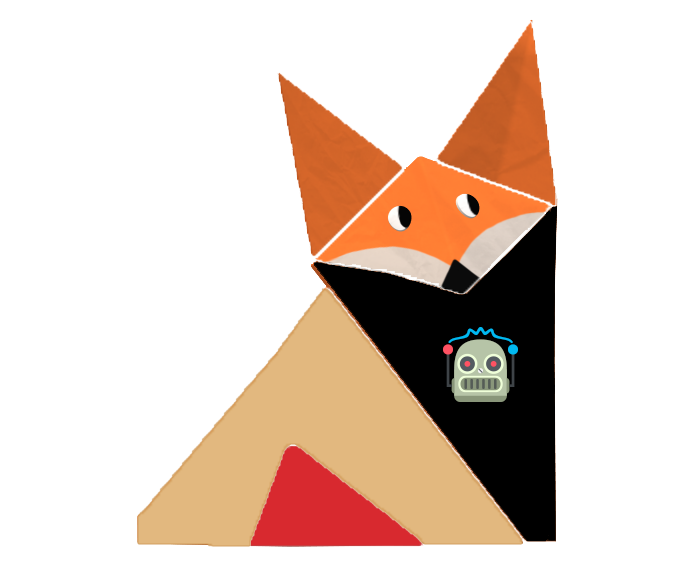
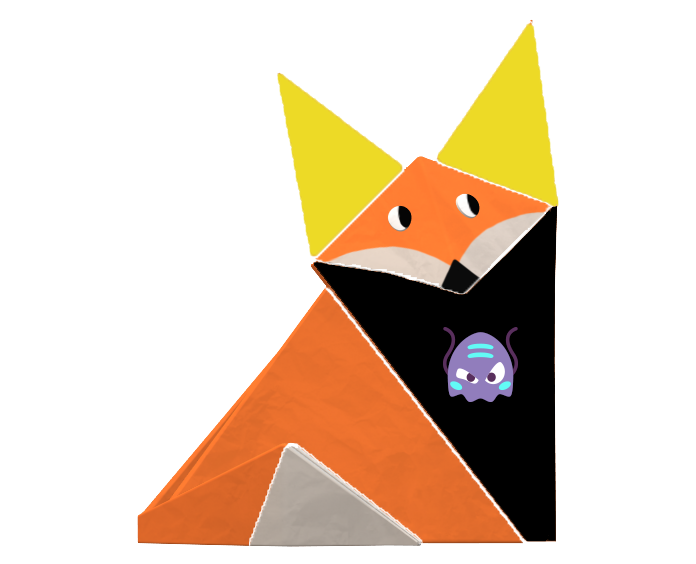
**PROTOTYPES (without emoji glyphs):**





**PROTOTYPES (with emoji glyphs):**



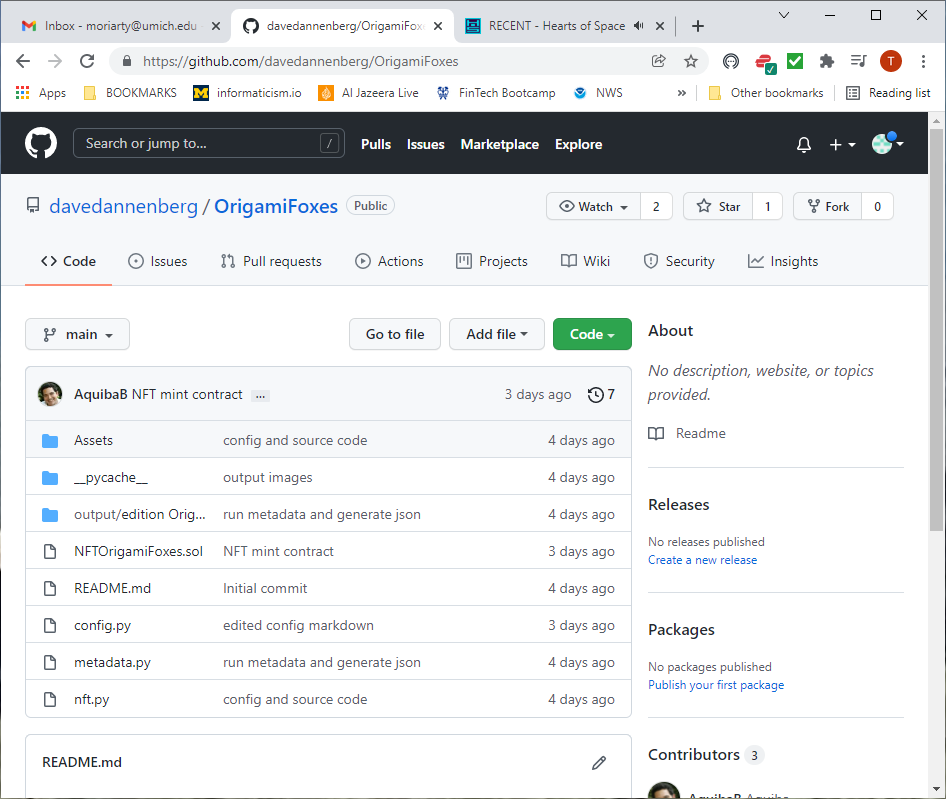


**config.py (note:** The following methodology has been co-opted from an article by Rounak Barik re: Scrappy Squirrels NFTs)

We implemented a Python list variable called CONFIG which contains a dictionary for each of our four trait categories, each with five keys: id, name, directory, required, and rarity\_weights.

CONFIG = [

{ 'id': 1,

'name': 'ear\_traits',

'directory': 'Ear',

'required': True,

'rarity\_weights': [20, 20, 20, 20, 20]},

{ 'id': 2,

'name': 'face\_tail\_traits',

'directory': 'FaceTail',

'required': True,

'rarity\_weights': [22, 22, 22, 22, 12]},

{ 'id': 3,

'name': 'lower\_body\_traits',

'directory': 'Body2',

'required': True,

'rarity\_weights': [18, 18, 18, 18, 18, 10]},

{ 'id': 4,

'name': 'upper\_body\_traits',

'directory': 'Body1',

'required': True,

'rarity\_weights': [5, 5, 5, 5, 16, 16, 16, 16, 16]},

]