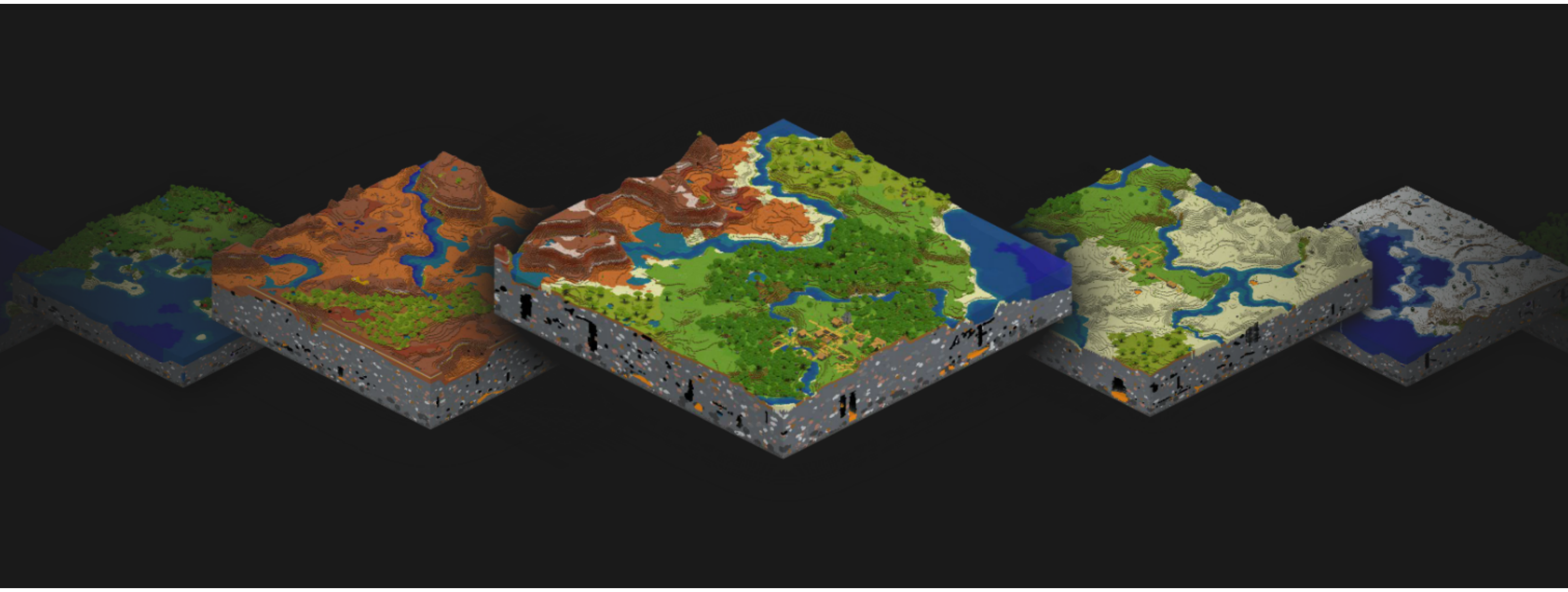
NFT Worlds

Launch Whitepaper

https://www.nftworlds.com/



What are NFT Worlds?

NFT Worlds are a collection of non-fungible tokens (NFTs) living on the Ethereum blockchain. (https://en.wikipedia.org/wiki/Non-fungible_token)

Each NFT World represents the idea of a singular virtual world. Each world has its own set of unique geography, resources, environments and special feature attributes stored immutably on the Ethereum blockchain.

Additionally, each NFT World can be virtually explored, developed and built in. Changes you make to an NFT World can then be synced to update the NFT's image to reflect the landscape changes and developments you've made.

Our Vision

We set out building NFT Worlds as an experiment to bring the idea of ownership backed, virtual land (or worlds) to the Ethereum blockchain as a platform.

There's a 2 sided audience to what we're doing, and compounding value we predict will be built over time by the community.

The first audience is developers and others that want to openly build with the on-chain data attached to each NFT World. Inspired by Loot, each world has its data on-chain, immutable, forever. We're excited to see others build interesting projects by using this data across all the worlds.

The other audience is creators. We built NFT Worlds to be compatible with Minecraft - using Minecraft specifically as a 3rd-party building and exploration tool to develop your world and play in it with others.

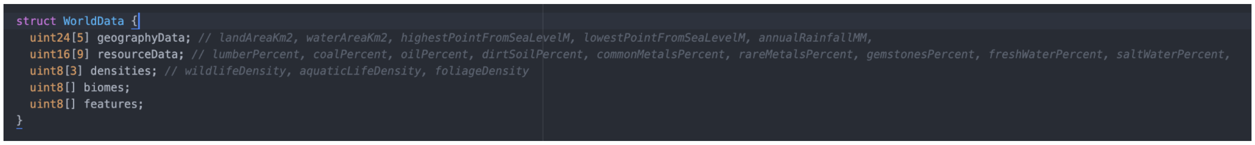
We see both developers & creators coming together in such a way over time that creates a blossoming ecosystem of NFT backed virtual worlds with extensive utility.

On Chain Data

All NFT Worlds are created with an extensive amount of on-chain data. This is submitted upon minting to be written to the blockchain. It is immutable and represents the baseline state of a world for other developers to create new layers on top of with dApps, tokens, etc.

Data payloads are signed with the contract owner's signature to prevent alteration of the intended data for each world upon minting.

NFT Worlds store their data on-chain in a struct as defined below.



```
struct WorldData {
  uint24[5] geographyData; // landAreaKm2, waterAreaKm2, highestPointFromSeaLevelM, lowestPointFromSeaLevelM, annualRainfallMM,
  uint16[9] resourceData; // lumberPercent, coalPercent, oilPercent, dirtSoilPercent, commonMetalsPercent, rareMetalsPercent, gemstonesPercent, freshWaterPercent, saltWaterPercent,
  uint8[3] densities; // wildlifeDensity, aquaticLifeDensity, foliageDensity
  uint8[] biomes;
  uint8[] features;
}
```

This struct represents the following world information.

Geography Data | uint24[5]
Values are stored as whole integers, to get their original % which should include 2 decimal places, simply divide the stored values by 100.

Use the **getGeography(_tokenId)** function of the contract to access this data for a given world id / tokenId (they are interchangeable)

- Index 0: Land area (kilometers²)
- Index 1: Water area (kilometers²)
- Index 2: Highest point from sea level (meters)
- Index 3: Lowest point from sea level (or deepest point in water/sea) (meters)
- Index 4: Annual Rainfall (millimeters)

Resource Data | uint16[9]
Values are stored as whole integers, to get their original % which should include 1 decimal place, simply divide the stored values by 10.

Each value represents a % value for a resource, relative to all total resources of the world. When you sum all resource values, this will equal 100%.

Use the **getResources(_tokenId)** function of the contract to access this data for a given world id / tokenId (they are interchangeable)

- Index 0: Lumber (%)
- Index 1: Coal (%)
- Index 2: Oil (%)
- Index 3: Dirt / Soil (%)
- Index 4: Common Rare Metals (%)
- Index 5: Rare Metals (%)
- Index 6: Gemstones (%)
- Index 7: Fresh Water (%)
- Index 8: Salt Water (%)

Densities | uint8[3] stored internally | string[3] returned externally
Values are stored as integers that represent internal lookup indexes, but the accessor method for this data will return an array of 3 strings.

Use the **getDensities(_tokenId)** function of the contract to access this data for a given world id / tokenId (they are interchangeable)

Possible values for each index are: "Very High", "High", "Medium", "Low", "Very Low".

- Index 0: The wildlife density of the world.
- Index 1: The aquatic life density of the world.
- Index 2: The foliage density of the world.

Biomes | uint8[] stored internally | string[] returned externally
Values are stored as integers that represent internal lookup indexes, but the accessor method for this data will return an array of strings, each string representing a unique biome/environment for this world. Will always return at least 1 string. Each string will always be unique relative to the others in the array.

Use the **getBiomes(_tokenId)** function of the contract to access this data for a given world id / tokenId (they are interchangeable)

Possible values of any index are: Forest, River, Swamp, Birch Forest, Savanna Plateau, Savanna, Beach, Desert, Plains, Desert Hills, Sunflower Glade, Gravel Strewn Mountains, Mountains, Wooded Mountains, Ocean, Deep Ocean, Swampy Hills, Evergreen Forest, Cursed Forest, Cold Ocean, Warm Ocean, Frozen Ocean, Stone Shore, Desert Lakes, Forest Of Flowers, Jungle, Badlands, Wooded Badlands Plateau, Evergreen Forest Mountains, Giant Evergreen Forest, Badlands Plateau, Dark Forest Hills, Snowy Tundra, Snowy Evergreen Forest, Frozen River, Snowy Beach, Snowy Mountains, Mushroom Shoreside Glades, Mushroom Glades, Frozen Fields, Bamboo Jungle, Destroyed Savanna, Eroded Badlands

Features | uint8[] stored internally | string[] returned externally

Values are stored as integers that represent internal lookup indexes, but the accessor method for this data will return an array of strings, each string representing a unique feature for this world. Will always return at least 1 string. Each string will always be unique relative to the others in the array.

Use the **getFeatures(_tokenId)** function of the contract to access this data for a given world id / tokenId (they are interchangeable)

Possible values of any index are: Ore Mine, Dark Trench, Ore Rich, Ancient Forest, Drought, Scarce Freshwater, Ironheart, Multi-Climate, Wild Cows, Snow, Mountains, Monsoons, Abundant Freshwater, Woodlands, Owls, Wild Horses, Monolith, Heavy Rains, Haunted, Salmon, Sunken City, Oil Fields, Dolphins, Sunken Ship, Town, Reefs, Deforestation, Deep Caverns, Aquatic Life Haven, Ancient Ocean, Sea Monsters, Buried Jems, Giant Squid, Cold Snaps, Icebergs, Witch's Hut, Heat Waves, Avalanches, Poisonous Bogs, Deep Water, Oasis, Jungle Ruins, Rains, Overgrowth, Wildflower Fields, Fishing Grounds, Fungus Patch, Vultures, Giant Spider Nests, Underground City, Calm Waters, Tropical Fish, Mushrooms, Large Lake, Pyramid, Rich Oil Veins, Cave Of Ancients, Island Volcano, Paydirt, Whales, Undersea Temple, City Beneath The Waves, Pirate's Grave, Wildlife Haven, Wild Bears, Rotting Earth, Blizzards, Cursed Wildlife, Lightning Strikes, Abundant Jewels, Dark Summoners, Never-Ending Winter, Bandit Camp, Vast Ocean, Shroom People, Holy River, Bird's Haven, Shapeshifters, Spawning Grounds, Fairies, Distorted Reality, Penguin Colonies, Heavenly Lights, Igloos, Arctic Pirates, Sunken Treasure, Witch Tales, Giant Ice Squid, Gold Veins, Polar Bears, Quicksand, Cats, Deadlands, Albino Llamas, Buried Treasure, Mermaids, Long Nights, Exile Camp, Octopus Colony, Chilled Caves, Dense Jungle, Spore Clouds, Will-O-Wisp's, Unending Clouds, Pandas, Hidden City Of Gold, Buried Idols, Thunder Storms, Abominable Snowmen, Floods, Centaurs, Walking Mushrooms, Scorched, Thunderstorms, Peaceful, Ancient Tunnel Network, Friendly Spirits, Giant Eagles, Catacombs, Temple Of Origin, World's Peak, Uninhabitable, Ancient Whales, Enchanted Earth, Kelp Overgrowth, Message In A Bottle, Ice Giants, Crypt Of Wisps, Underworld Passage, Eskimo Settlers, Dragons, Gold Rush, Fountain Of Aging, Haunted Manor, Holy, Kraken

World Renders, Building & Exploration

World Renders & How It's Done
All NFT Worlds were generated using an open source world generator that can take a number or "Seed" to generate world data that is compatible with Minecraft, or backwards compatible by inputting the same seed into the Minecraft client to get a copy of the world represented on your NFT to play/build in. (Compatible & tested with Minecraft Java edition 1.17, likely compatible with bedrock & other pc versions)

These data files were then patched into an isometric renderer tool that we developed internally.

We also use our own complete tileset for these isometric renders.

This is how we get the unique imagery of each world on the NFT's, and how we can perform future isometric render of world data files from a world to sync your changes back to your NFT.

Exploring, Building, Syncing What You Build Back To Your NFT
Each NFT has a unique seed that can be accessed through the **getSeed(_tokenId)** function of the contract. This function can only be accessed by the NFT's owner.

You can use Minecraft as a tool to explore, build & share these worlds with others. Upon creating a new world, enter your world's unique seed into the world creation "seed" field of Minecraft to enter your world.

When you've built something you're excited about and want to sync it back to your nft, you'll be available to access a dApp at https://www.nftworlds.com/sync (This will be available within 6 weeks after minting) where you can upload your world data file & submit a

corresponding transaction against the NFT World's smart contract to update your world's image to reflect your changes.
Additionally, your world data will be uploaded to IPFS (https://en.wikipedia.org/wiki/InterPlanetary_File_System) & can be made available for others to see & play, or kept private for only you or future owners of your NFT to access.

Verified Smart Contract

https://etherscan.io/address/0xBD4455dA5929D5639EE098ABFaa3241e9ae111Af

Disclaimers

NFT Worlds is in no way associated with, endorsed or a partner of Minecraft, Mojang, Microsoft or any related parties.

NFT Worlds were released for free to initial "minters" on October 5th, 2021 at 10PM PST.

NFT Worlds believes it falls within the bounds of the Minecraft EULA as transformative fair use.