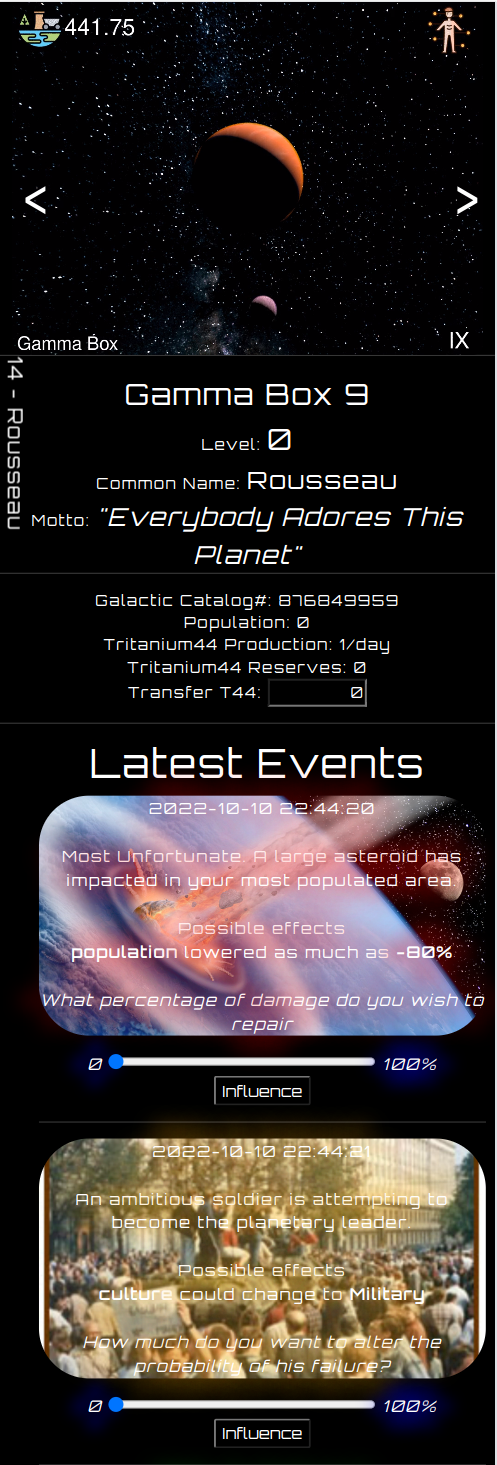
Kaon Planet

1. The virtual Trillionaire idle pastime of discovering, evolving and selling planets and their products.
   1. The player is “The God” of the planet, the manipulator who works behind the scenes to guide the population of the planet to improve their tech level and Tritanium44 production.
   2. Kaon Planet is a game for people who don't have time to sit in front of a computer screen all day long, but like to engage in strategic games or pastimes.
   3. All that is required to play the game is owning one (or more) of the Kaon Planets, and a cell phone or computer.
   4. The only participation required is once an hour events will be presented on the game screen which the player may, or may not decide to support or oppose, by simple clicks of buttons.
2. The game is actually 4 separate activities rolling into one:
   1. NFTs - Each planet in the game is an actual NFT on either Polygon, Ethereum, Avalanche, Fantom, Cronos or Solana blockchain.
   2. Animations - The Kaon Planets are not just static images that sit there. They are animations that are NFTs, but they can also be evolved and modified into additional NFTs as the owner adds resources or plays the Kaon Planet game well
   3. The NFT itself is always owned lock,stock and barrel by the player, like any other NFT. However, the civilization on that planet is owned by the Kaon Game. The rewards in Tritanium44 are handed out by the Kaon Game based upon how well the player develops the planetary
3. Tritanium44 (T44) is the crypto token that is the reward for playing the game well. It is an ERC20 token owned lock, stock and barrel by the player.
   1. The Kaon Game can only reward Tritanium44 to the Planetary Treasury.
   2. It is the player who moves that T44 from the Planetary Treasury, into actual T44 on the blockchain. The Kaon Game will not, nor is it capable, of taking T44 back from the player’s wallet.
   3. T44 can be exchanged for any other ERC20 token on exchanges that support it, or DEX (Distributed Exchanges) that support it.
   4. The amount of T44 each Kaon Planet collects each day is determined by:
      1. The base earning of any Kaon Planet is 1 T44 per hour.
      2. The Tech Level of the planet which the player can raise by making the right moves in the game play can greatly increase the reward
      3. The planetary population can greatly increase the reward
      4. Collecting Kaon Planets that are all of a particular set will increase the reward. For example, if ALL of the planets of a solar system are collected, 1 thru 12, this doubles the Tritanium44 rewards for each of the planets on top of any other rewards multipliers. Collecting somewhat fewer than all the planets also increases reward
      5. Having an entire set of each of the lifeforms doubles the rewards of the planets in that set.
      6. Having an entire set of each of the life chemistries doubles the rewards of planets in that set.
      7. Trading
         1. Upgrades to planets require increasing tech levels and acquiring the materials, infrastructure and manufacturing level to support those upgrades.
            1. Upgrades include new infrastructure to increase the population and higher tech levels (schools, cities, orbital platforms, space elevators etc)
         2. Formulas to make refined or manufactured goods, which in turn can be used to make further refined or manufactured goods and further to make the planetary upgrades.
         3. All transactions between players must be done with Tritanium44 (which will start to give it actual value due to demand)
         4. Moons and rings contain raw materials that the player will not know about until reaching certain tech levels. They are not unlocked to be usable until the planet achieves space based industry.
4. Overall Principles
   1. The game is currently only player vs. environment, not player vs player. Player vs player will only ever be optional. Casual players should feel comfortable just getting a message every hour about how their planet is doing, and making a few minor decisions about the direction of its evolution.
5. Game Play
   1. The game is set up so that every hour a player is presented with some events that have occurred, and probably decisions to make about the planet. For example: a plague breaks out, how much do you want to spend to lower the damage it does? An ancient high tech relic is discovered, how much do you want to invest in figuring out its technology?
   2. Every hour the player earns at least 1 Tritanium44. By collecting planet sets, raising tech level, and population or trading with other players, this hourly production can be vastly increased.
   3. Every hour the player has an entire hour to make, or re-make the moves he wants, then the “resolver” process runs and takes all the moves the players have registered and the new status of affairs is presented on the User Interface.
   4. Planetary cultures affect how tech levels progress and the probability for various types of event increases and decreases in accordance. For example, wars will occur much more often with Military cultures. Egalitarian cultures make much faster progress but are subject to problems of their own. Some life forms need certain types of cultures to make progress. Certain life forms are immune to certain types of disasters. For example, gaseous life forms are immune to quakes. But these life forms make very little progress with material sciences (very little ability to manipulate the environment). There are all sorts of trade offs and algorithms dealing this this yet to be worked out.
   5. Cultures can exchange knowledge and infrastructure. For example, a gaseous life form planet can purchase hard mining equipment from another culture in exchange for wisdom it has worked out about how cultures evolve.
6. User Interface
   1. The UI is one screen usable on a mobile device. Different sections are access by scrolling up or down.

Upper section is the animated NFT showing the total resources of the planet (441,75), the dominant indigenous life form (bipedal humanoid) and the official designation (Gamma Box IX).

The left and right arrows switch between multiple planets that the player owns

(The sideways text scrolls to be always visible so when looking at lower portions of the screen, the player knows which planet this is)

Level is the average of all the tech levels. This is the Equivalent to a Level in an typical video game.

Common name and motto are things the owner can set to add some of their own personality to the planet.

The Galactic Catalog number is the key which holds all the data about this planet together in the back end.

Tritanium44 production is how many T44s the player gets each turn (hour long).

Reserves is how much is in the Planetary Treasury (to use the T44 for the game, the T44 must come from the Treasury)

T44 can be transferred to and from your crypto wallet here.

Latest Events are where the player is confronted by situations to deal with. How these are handled will determine if his planet goes up in tech levels, or descends back to 0. The game requires active, human attention to create Tritanium44 which is why this form of mining is called, “Proof by Attention”.

1. Planets
   1. Each planet is an NFT, animated mp4 file and accompanying .png still image. (The .png is just the extracted first frame of the mp4)
   2. Planets have certain attributes. This data is stored in the Moralis database but ALSO as JSON right in the MetaData of the mp4 file.
   3. This base information cannot be changed. A planets qualities are set in stone for the purposes of the game
      1. Administrative Attributes
         1. planet designation – Star Index, Star System and Planet Index (Star Index is which star in the start system, and Planet Index is which planet out of the 12 planets surrounding each star)
         2. evaluation – The primary opinion the Galactic Confederation has about the planetary
         3. evaluation2 – the secondary opinion the GF has about the planet.
         4. lifeform – the shape of the dominant, indigenous life form. 2 legs? 4 legs? Aquatic? Plant? Reptile etc.
         5. chemistry – The chemical basis of the life, This is just the column names from the periodic table (families, groups of elements) (Halides, Halogens, etc)
         6. planet\_type – This is just the base name of the planet surface map. If the surface map is terrestrial\_24.jpg, the planet is “terrestrial” type. Other types are oceanic (mostly ocean), aquatic (all ocean), terrestrial (some ocean, some land), desert (all land, no oceans and almost no liquid) forest (all land covered with plants), ice (completely covered with ice) glaciated (lots of ice, but some land and ocean). Note oceans are not necessarily water oceans, they could be methane, bromine, ammonia or other chemicals. Land however, it probably mostly rock which is pretty common in the universe.
         7. unexplored – This must means that there is no writing on the planet mp4. This makes is possibly more usable as a desktop background or image to display in a art gallery.
         8. Artist - the person which generated the planet (or anonymous)
      2. Planetary Resources – The raw materials available on this planet (in EEM “Earth Equivalent Mass”)
         1. rare – rare ores and minerals
         2. radioactives – minerals with radioactive aspects needed to initiate nuclear reactions
         3. refractories – high temperature metals needed for high stress, high temperature applications
         4. industrials – common materials used in large quantity for infrastructure and mass produced products
         5. biologicals – life and life byproducts
         6. exotics – materials only available form this planet, like “spice” of Dune, (also a biological), or “thionite” from Trenco (also a biological)
         7. relics – artifacts from previous or alien cultures that, if analyzed properly, can result in large tech level jumps
2. Cultures
   1. Cultures are what the game play revolves around
   2. The aspects of the culture can be modified by the player but only when opportunities arise in the events. Modifying the culture type for example can only occur when a revolutionary is born, raising a tech level can only occur if a breakthrough engineer, scientist or philsopher is born, or perhaps some other events give the society a boost up.
   3. Other aspects such as built up infrastructure will likely not be included in the first release of the game, but maybe they will. It depends how the play testing goes.
3. Trading Contracts (not implemented yet)
   1. Trading within the same Solar System requires interplanetary industry tech level
   2. Trading outside Solar System requires interstellar tech level
   3. Not implemented at all but the idea here setting up automated agreements with other players so that each turn X amount of Tritanium44 is exchanged for Y amount of some commodity. These commodities are needed to fuel development of your planet. This won’t be implemented right away for the purpose of keeping the game simple for now.
   4. Each contract will have the Other Planet in the deal, the amount of Tritanium exchanged, and the materials exchanged.
   5. The contracts run eternally until either party cancels or a failure to deliver occurs on either side. There is no requirement to stay in a contract though it may piss off your trading partner if he was depending on the materials you were sending.
4. Developments (not implemented yet)
   1. This is where the materials you acquire are used to create other materials or projects. Some projects don’t require multiple turns but some are continuous and others require several turns to complete.
   2. Improving your planet requires creating assets such as schools, factories, cities etc to increase the production of Tritanium44
5. Developer Notes
   1. The planets are created using the Planet Generator. This is a UI that generates planets randomly, then lets the user tweak the results.
   2. Deploying the planets as NFTs is a somewhat manual process supported by commandline tools.
   3. The back end database is kept on Moralis, a Web3 infrastructure library and company started by Ivan on Tech which makes Web3 easier. Using the same functions, it supports all the popular Solidity/ERC20 type blockchains.
   4. The “Events” that planets get hit with are simple JSON statements like this:
   5. Planets are generated using a commandline/config file driven, ray tracing image generator called Povray. <https://www.povray.org/>.
      1. Why was this chosen?
      2. Originally the game was envisioned as a way for people to mint their own planets (this is still on the table for the future) and this required a website user interface where users specified aspects of the planet then an automated back end would generate them. Most, if not all graphics program have a highly detailed, and complex visual front end, some of the most complex in all computer software. Not really good for NFT buyers who aren’t necessarily graphics experts. So a simplified front end was created (and still exists but only for use by game designers) where the user can only really randomly generate the planets, then specify certain aspects to be regenerated (“roll the dice again for the atmopshere” if you don’t like that.) like creating a Dungeon Character)
      3. In the end, generating these planets is a very CPU intense process (100 high resolution images are stitched together to make the animations) and if someone is sitting there making tons of animations, tweaking them, just to get it perfect, the ability to support many users doing this at the same time is very limited without a massive back end.
      4. Visual Attributes are what control the planet creation. Unless specified, each attribute is randomly generated. If using the Web User Interface then each iteration of the planet can be tweaked and one still regenerated. When the creator is satisfied with the look of the still image, Create Planet will create the full animation.
      5. The commandline version just accepts parameters on the commanline, like FRAMES=100 ./planetor.py camera\_location=10,200,30 sun\_position=1000,-1000,1000 planet=terrestrial\_10.jpg clouds=clouds\_2.png … etc etc.
      6. ./planetory.py will generate as many frames as specified into an animation. The 100 high quality images are generated then passed to the generate.sh script which is a shell script that runs several graphics processing utilities to add background, stitch together the 100 images into an mp4 animation (and optimize it) and add visual, written content to the mp4.
         1. There are some visual attributes that are fed into the Planet Maker (or generated randomly by the Planet maker if not explicitly stated)
            1. identity – a random number generated when the planet is first created. All subsequent information about the planet is linked together using this id.
            2. scene – A template POVRAY .ini file that contains variables that get filled out the User Interface choices.
            3. camera\_location – Where in the scene the viewer is positioned
            4. camera\_angle – the angle of the viewer
            5. camera\_look\_at – where is the camera looking
            6. sun\_brightness
            7. sun\_position
            8. planet\_size
            9. background – backgrounds of stars or nebulae
            10. planet – a texture map of the planetary surface
            11. clouds – a texture map of atmopsheric clouds
            12. clouds\_density – how transparent the clouds are
            13. atmosphere – color of the thin layer of gases surrounding the planet
            14. atmosphere\_density – How transparent the atmosphere is
            15. moons – How many moons does the planet have
            16. rings – How many rings does the planet have