$CAPYBARA — From evaluation to rewards and skills exchange

🦫 Overview

Capybara is a decentralized ecosystem designed for families and children, providing a gamified platform to stimulate learning, collaboration, and economic empowerment. Leveraging Gate NFTs, $CAPYBARA tokens, and secure family settings stored on IPFS, the project offers a unique approach to financial education and interaction.

🎯 Goals

1. **Financial Literacy**
   Teach children to handle digital tokens responsibly, reinforcing concepts of honesty and mutual respect through everyday $CAPYBARA interactions.
2. **Family as a Micro-Economy**
   Enable families to model economic systems by managing stablecoins, distributing $CAPYBARA for meaningful tasks, and fostering a sense of shared values.
3. **Encouraging Collaboration**
   Build an ecosystem where children create and trade within their network, fostering creativity, self-reliance, and community support reducing their dependence on “adult economy”.

📜 Smart Contracts

CapybaraToken.sol

1. **Token Standard**
   - The `CapybaraToken` is an ERC-20 token implementation built on OpenZeppelin's ERC-20 library.
   - It follows the standard functionalities of a fungible token, including transfers, balance queries, and approvals.
2. **Initial Supply**
   - Upon deployment, the contract mints a fixed initial supply of `100 * 10^18` tokens (100 tokens with 18 decimals) and assigns them to the owner specified during deployment.
   - The constant `INITIAL_SUPPLY` ensures transparency about the initial token allocation.

3. **Minting and Ownership**
    - The `Ownable` module from OpenZeppelin is used to restrict critical functionalities (like minting new tokens) to the contract owner.
    - The constructor sets the initial owner, ensuring that tokens and privileges are securely assigned.
4. **Constructor Logic**
    - Ensures that:
        - The `owner` address is not zero.
        - The `INITIAL_SUPPLY` is greater than zero.
    - Transfers ownership to the specified `owner` and mints the initial supply to this address.
5. **Usage Context**
    - This token serves as the core currency (`$CAPYBARA`) within the Capybara ecosystem.
    - It is used for:
        - Rewarding achievements and tasks.
        - Facilitating token exchanges within families.
        - Supporting the ecosystem's gamified interactions.
6. **Scalability**
    - The contract is simple and does not currently include dynamic minting or burning logic, focusing on a fixed supply model.
    - Future upgrades could introduce additional functionalities like staking or burning mechanisms.

CapybaraNFTGate.sol

NFT Minting
The `CapybaraNFTGate` contract enables minting of Gate NFTs through the `mintGateNFT` function. Each NFT is associated with metadata stored on IPFS and includes a tariff that specifies the amount of `$CAPYBARA` tokens awarded to the buyer. When a new NFT is minted, its ownership is tracked, and a `GateNFTMinted` event is emitted.

Ownership Management
Ownership of NFTs is tracked using mappings that store the tokens owned by each address. These mappings enable efficient enumeration and transfer of tokens. When a token is minted or transferred, its ownership records are updated to reflect the change. This ensures precise tracking of token holders without the need for computationally expensive loops.

Family Settings and Admin Management
Each Gate NFT is linked to a family through a unique family ID. Family settings, including IPFS CIDs for configuration data, are stored and managed dynamically. Administrative access for managing family settings is granted based on the family ID, allowing specific addresses to make updates while ensuring security and transparency.

Interaction with External Contracts
The contract integrates seamlessly with `CapybaraToken` and `CapybaraDispenser`. It

uses `CapybaraToken` to reward buyers with `$CAPYBARA` tokens during NFT sales. Additionally, it interacts with `CapybaraDispenser` to manage family liquidity, transaction limits, and other parameters based on Gate NFT configurations.

Family Creation and Updates
The `onNFTSold` function is triggered when an NFT is sold, creating a unique family ID for the buyer and linking it to the NFT. This function updates family settings, assigns administrative rights to the buyer, and rewards them with `$CAPYBARA` tokens. Admins can later modify family settings using the `updateFamilySettings` function, which updates data on IPFS and adjusts configurations in `CapybaraDispenser`.

Custom Query Functions
The contract includes helper functions such as `getOwnedGateNFTs`, which returns a list of all NFTs owned by a specific address. This functionality simplifies tracking ownership for both users and the ecosystem.

Token URI Management
Each NFT's metadata is linked to a unique URI stored on IPFS. The contract allows for easy retrieval and updating of these URIs, ensuring that token metadata remains accurate and up-to-date.

Event Emissions
The contract emits events to signal key actions, such as NFT minting, family creation, and token rewards. These events enable off-chain systems and users to monitor activities within the Capybara ecosystem in real time.

CapybaraDispenser.sol

Family and Token Management
The `CapybaraDispenser` manages family-specific settings, rewards, and token exchanges within the Capybara ecosystem. It associates each Gate NFT with a unique `familyId`, enabling detailed customization of liquidity, transaction limits, and other parameters. The contract also tracks family administrators, children, and their economic activities.

Liquidity Management
Family-specific liquidity is tracked and updated using stablecoins. Administrators can deposit liquidity into the system, which is then utilized for token exchanges and rewards distribution. Liquidity is tied to the family and the stablecoin defined in its settings.

Rewards and Exchange Mechanism
The contract facilitates the exchange of `$CAPYBARA` tokens for stablecoins based on a globally defined exchange rate. It also allows families to earn DOGE rewards through a structured activity index system. Rewards are calculated and distributed based on family and global activity indices.

Integration with Gate NFTs
Gate NFTs are used as the entry point for creating family settings. Upon initialization or sale

of a Gate NFT, the dispenser updates family configurations, linking them to metadata stored on IPFS. This ensures that families are dynamically tied to their Gate NFTs.

Activity Tracking
The contract tracks activity at both the family and global levels. These indices are used to monitor interactions within the ecosystem and influence token rewards and DOGE distribution. Activities are logged for each family member, ensuring accountability and transparency.

External Service Interaction
The dispenser integrates with an external service to fetch and manage metadata linked to Gate NFTs. This enables seamless interaction with IPFS for storing and retrieving family configuration data.

Administrative Functions
Administrators of a family can:

- Update family settings, including liquidity, transaction limits, and rewards distribution.
- Monitor and manage the activity index for both children and families.
- Assign new administrators and children within their family.

Custom Query and Utility Functions
The contract includes utility functions to:

- Check if an address belongs to a child or adult role.
- Retrieve family settings and roles linked to a specific address.
- Access stablecoin and liquidity distribution metrics for each family.

Event Emissions
Key events provide transparency and enable off-chain integrations:

- `MetadataFetched`: Signals the retrieval of metadata from IPFS.
- `FamilySettingsUpdated`: Indicates changes in family configurations.
- `LiquidityDeposited`: Logs liquidity additions for families.
- `CapybaraExchanged`: Tracks token exchanges for families and children.
- `ChildActivityUpdated`: Monitors activity index changes for family members.
- `DogeRewardsDistributed`: Logs DOGE reward distributions based on activity.
    -

🧪 Testing Scenarios

1. **Minting and Selling Gate NFTs**
    - Simulate NFT sales on a local blockchain followed by setting-up the family preferences.
    - Ensure the buyer receives $CAPYBARA tokens for the selected “price plan”.

    - Validate positive and negative scenarios for access management governed by purchasing multiple Gate NFTs accumulating available for the family tokens volume. The pricing evolution must be non-linear in order to avoid potential complaints to fiat mapping (similar to stablecoins that need a MiCa regulation).
    - Validate access control for family administrators and children addresses linked per family for all public actions.
2. **IPFS Integration**
    - Verify automatic JSON generation for family settings.
    - Test updating family data with new CID.
3. **Dispenser Functionality**
    - Validate an entire cycle for the token (reward) → child, child (gets liquidity for the effort) → dispenser→ stablecoins, parent (refresh liquidity pool) → dispenser → family pool update

---

🌍 Integrations

- **Marketplaces:** Compatible with Estonians888 Marketplace (WIP) for decentralized NFT sales.
- **Wallets:** Fully interoperable with MetaMask via “custom tokens” feature.

🚀 Current state And Future Enhancements

Token is deployed and used in the context of my family:
0x5b014A5fe7E970E9B4dF2c4EF8b2b0c8537e4568

Contracts for Dispenser and NFT Gate are developed and pushed but not yet tested locally with a full cycle and not yet deployed. After that a simple Web UI can be integrated with MetaMask in order to provide liquidity for Dispenser and update family settings for Gate NFTs. Children can use Metamask on the phone, works pretty well if their parents are in IT :). For wider audience we can create a fancy DApp with a playful child oriented design and most probably more features oriented to stimulate child-2-child co-creation and values exchange.

Contact me on Telegram (@zeya888) if you feel like you want to contribute or be among the first free families. Monetisation for contributors will come from NFT Gate sales that identify a tier of liquidity volume. Or course I would love to make it a DAO and let it be in the hands of responsible caring parents 💜.