




NFT Pets



By: The Quagga Group
Ethan Silvas, Naomi Velasco, Karim Bouzina, and Jeff Crabill



Mission Statement

Create NFT pets out of DALL-E generated pixel art images with functionality to chat with them via ChatGPT and buy/sell them on an Ethereum marketplace.



Project Goals

- Build on what we learned about ERC721 tokens by creating an NFT marketplace
- Leverage OpenAI APIs to integrate AI tools into a blockchain app
- Combine the blockchain backend and Python Streamlit frontend to create full-stack dApp

Project Summary

- Create a Solidity smart contract to create ERC721 tokens
 - Add functionality to buy and sell tokens between Ethereum addresses
 - Implement “middleman” to take 2% cut of every transaction
- Use DALL-E API to generate unique pixel art NFT pets
- Use ChatGPT API to create chat interactions between NFT pets
- Put all of these elements together in a Streamlit UI with Metamask, Ganache, and Web3.py



Login

Logged in

Logout

[Register New Pet](#) [Chat](#) [Your Pets](#) [Marketplace](#)

Chat With Your NFT Pets

Select a Pet

barry



Use the text box to send a message to your pet:

Building From ArtToken

```
// function similar to ArtToken registerArtwork; emits PetRegistered event
function registerPet(string memory name, string memory ownerName, uint256 price, string memory tokenURI, bool isBuyable) public returns (uint256) {
    uint256 tokenId = totalSupply();
    _mint(msg.sender, tokenId);
    _setTokenURI(tokenId, tokenURI);

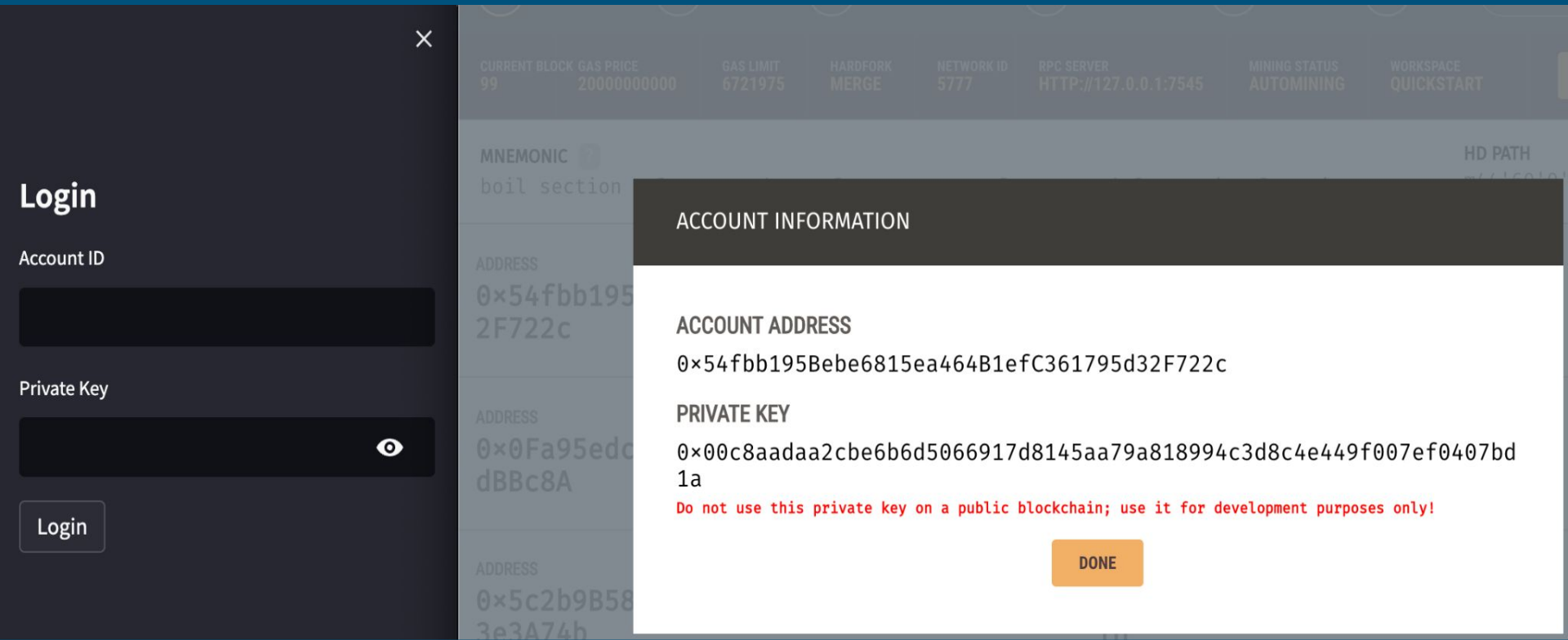
    pets[tokenId] = Pet(name, ownerName, price, isBuyable);

    emit PetRegistered(tokenId, name, ownerName, price, isBuyable, middleman);

    return tokenId;
}
```

- Similar base logic to ArtToken
- Use events to efficiently transfer information

Add Login Capabilities With Ganache



Register New NFT Pet

Select animal

Alligator

Name

Price in Wei

List Pet on Marketplace?



Yes



No

Register NFT Pet

- Register pets to logged in account
- Add option to list on marketplace


```
// events to track minting a pet, changing the price of a pet, and the sale of a pet
event PetRegistered(uint256 tokenId, string name, string ownerName, uint256 price, bool isBuyable, address indexed middleman);
event PetPriceChanged(uint256 tokenId, uint256 price);
event PetSold(uint256 tokenId, address oldOwner, address newOwner, uint256 price);
```

```
// transfer the 2% fee to the middleman address
uint256 fee = price * 2 / 100;
middleman.transfer(fee);

_transferFrom(oldOwner, newOwner, tokenId);

// change owner name within Pet struct
pets[tokenId].ownerName = ERC721Metadata(address(this)).name();

oldOwner.transfer(price - fee);

emit PetSold(tokenId, oldOwner, newOwner, price);
```

NFT Pet Marketplace



Name: Samson

Price: 1000 Wei

Buy Pet



Name: Doug

Price: 100000000 Wei

Buy Pet

Use buyPet() to Create Marketplace

- 2% goes to address that deployed the smart contract
- Ownership is immediately reflected after purchase

Chat With Your NFT Pets

Select a Pet

Samson



Use the text box to send a message to your pet:

Hi Samson

Hi Samson

Samson: Hello! I'm not sure if I'm an actual NFT pet or if you're just using that as a metaphor, but either way, how can I assist you today?

ChatGPT to Interact With Pets

- Chat will continuously add to conversation
- Can kind of give it a “personality”

OpenAI: DALL-E and ChatGPT APIs

- Only a few lines of code!
- Pay a small fee to generate images and interact with chat
- Relatively fast response times

```
# create DALLE prompt based on chosen animal - ex: "pixel art dog"
animal = str(selection.lower().replace(' ', '_'))
PROMPT = (f"pixel art {animal}")

# call openai dalle api to generate image
def generate_nft_pet():
    response = openai.Image.create(
        prompt=PROMPT,
        n=1,
        size="256x256")
    return response
```

```
response = openai.ChatCompletion.create(
    model='gpt-3.5-turbo',
    messages=[
        {"role": "system", "content": "You are a NFT pet."},
        {"role": "user", "content": text},
    ]
)
```

Results

- Successfully created marketplace transactions that were managed by UI and smart contract
- Integrated OpenAI APIs for out of the box AI tools
- Created a dApp with a frontend that combines the blockchain backend and AI tools

Issues

- Not 100% ready to deploy as an actual app
 - UI bugs
 - Needs Metamask login/logout integration
- DALL-E images aren't saved forever and enabling them to do so might cost more
- Web3.js offers more support than Web3.py

Next Steps

- Deploy the app
 - Convert to JavaScript/React.js UI instead of Streamlit
 - Implement Metamask login/logout functionality for better security
- Add more marketplace features
 - Modify your listings
 - See your sales and profit history
- Implement Pokemon-like game features

Thank You!
