ERC721开发实战

蔡一@志顶科技

2018.03.03

目录

谜恋猫CryptoKitties

让以太坊拥堵的撸猫游戏

可收藏
可繁殖
讨人喜欢

收藏并繁殖数字喵咪

开始游戏

待收养　待交配　零代猫　所有猫咪

排序　高价猫咪优先

118,032 猫咪

筛选猫咪

猫咪 193760 · 6 代 · 快速

16

猫咪 58177 · 5 代 · 快速

15

猫咪 151911 · 6 代 · 中速

15

猫咪 193659 · 15 代 · 慢速

23

猫咪 160999 · 5 代 · 快速

27

猫咪 26171 · 2 代 · 中速

86

猫咪 148645 · 4 代 · 急速

猫咪 250478 · 19 代 · 迟缓

7

待收养 待交配 零代猫 所有猫咪

排序 高价猫咪优先

33,690 猫咪

筛选猫咪

猫咪 382351 · 5 代 · 快速

34

猫咪 25763 · 12 代 · 迟缓

8

猫咪 503717 · 11 代 · 中速

0

猫咪 438352 · 14 代 · 慢速

2

猫咪 288572 · 14 代 · 迟缓

1

猫咪 91040 · 16 代 · 迟缓

1

猫咪 230245 · 17 代 · 迟缓

猫咪 84978 · 17 代 · 龟速

2

待收养 待交配 零代猫 所有猫咪

排序 高价猫咪优先

1,098 猫咪

筛选猫咪

新的“零代”猫咪大约每15分钟产生一只！

查看最新猫咪!

猫咪 430720 · 0 代 · 神速

11

猫咪 117911 · 0 代 · 神速

12

猫咪 118472 · 0 代 · 中速

5

猫咪 110005 · 0 代 · 急速

0

游戏规则

- 销售总量50,000个，每只具有独一无二的外形
- 不仅是一种数字化藏品，并且可通过交配繁育后代
- 猫的繁殖是有冷却时间，冷却时间代际递增
- 零代猫采用递减拍卖的形式销售
- 零代猫的初始价为最近5只成交均价的1.5倍
- 玩家可在市场出售自己的猫
- 玩家也可在市场上出售猫的交配权
- 玩家出售的方式也是递减拍卖

背后的协议：ERC721

ERC721： Non-fungible Tokens (NFTs) Standard

ERC721：针对非标通证的标准接口

非标通证（NFTs）

每个通证都是独一无二的

标准通证

- ERC20（可互换的）
- 每个通证的价值都是相同的，类似于现金的概念，你无须在意收到的是哪一个通证
- 区块链管理的是通证的数量，而不是某个特定的通证
- 比如股票和货币

非标通证

- ERC721（不可互换的）
- 每个通证都是独一无二的，价值都是不同的，类似于数字化藏品的概念
- 区块链需要管理每个通证
- 比如CryptoKitties

ERC 721接口（一）

```
function totalSupply() public view returns (uint256 total);
function balanceOf(address _owner) public view returns (uint256 balance);
function ownerOf(uint256 _tokenId) external view returns (address owner);
function approve(address _to, uint256 _tokenId) external;
function transfer(address _to, uint256 _tokenId) external;
function transferFrom(address _from, address _to, uint256 _tokenId) external;
// Events
event Transfer(address from, address to, uint256 tokenId);
event Approval(address owner, address approved, uint256 tokenId);
```

https://github.com/ethereum/EIPs/issues/721

ERC 721接口（二）

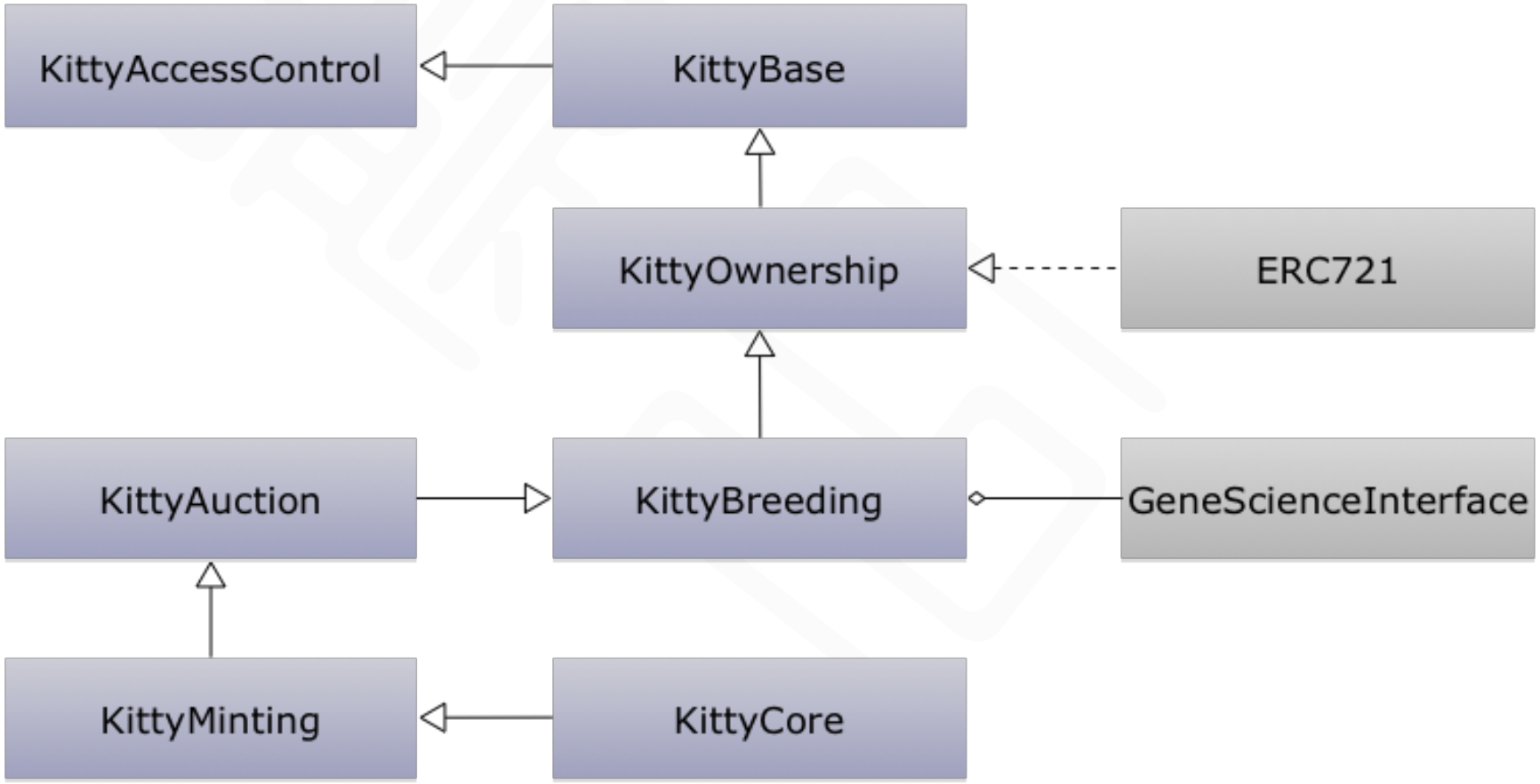
```
// Optional
function name() public view returns (string name);
function symbol() public view returns (string symbol);
function tokensOfOwner(address _owner) external view returns (uint256[] tokenIds);
function tokenMetadata(uint256 _tokenId, string _preferredTransport) public view
returns (string infoUrl);

// ERC-165 Compatibility (https://github.com/ethereum/EIPs/issues/165)
function supportsInterface(bytes4 _interfaceID) external view returns (bool);
```

https://github.com/ethereum/EIPs/issues/721

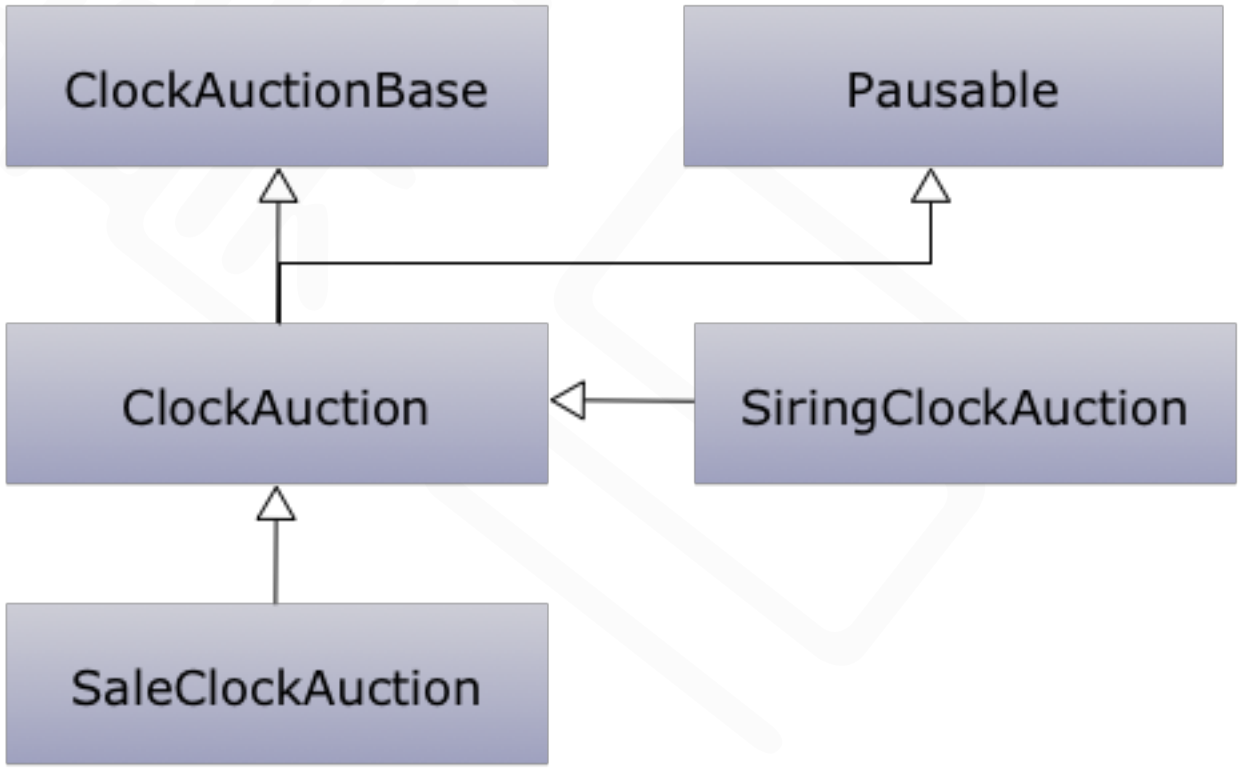
他山之石

CryptoKitties代码分析

CryptoKitties代码分析—Kitty



CryptoKitties代码分析—Auction



金鱼品种大全

开发自己的ERC721合约

金鱼品种大全—形态特征

- 体色
- 头型
- 鳞片
- 体型
- 体纹
- 尾巴
- 眼睛

金鱼品种大全—FishBase-1

```
contract FishBase {
    event Transfer(address from, address to, uint256 tokenId);
    struct Fish {
       uint64 birthTime;
       uint256 genes;
       uint256 price;
    }
    Fish[] fishes;
    mapping (uint256 => address) public fishIndexToOwner;
    mapping (address => uint256) ownershipTokenCount;
```

金鱼品种大全—FishBase-2

```
function _transfer(address _from, address _to, uint256 _tokenId) internal {
    if (_from != address(0)) {
            ownershipTokenCount[_from]--;
     }
     ownershipTokenCount[_to]++;
     fishIndexToOwner[_tokenId] = _to;

     Transfer(_from, _to, _tokenId);
}
```

金鱼品种大全—FishBase-3

```
 function _createFish(uint256 _genes, uint256 _price, address _owner) internal
returns (uint) {
        require(_owner != address(0));

        Fish memory _fish = Fish({
          genes: _genes,
          birthTime: uint16(now),
          price: _price
      });

        uint256 newFishId = fishes.push(_fish) - 1;
        _transfer(0, _owner, newFishId);

        return newFishId;
   }
```

金鱼品种大全—FishCore-1

```
contract FishCore is FishBase, ERC721 {
    string public constant name = "GoldFish";
    string public constant symbol = "GF";
    function _owns(address _claimant, uint256 _tokenId) internal view returns (bool) {
        return fishIndexToOwner[_tokenId] == _claimant;
    }
    function balanceOf(address _owner) public view returns (uint256 count) {
        return ownershipTokenCount[_owner];
    }
```

金鱼品种大全—FishCore-2

```
function transfer(address _to, uint256 _tokenId) external {
      require(_to != address(0));
      require(_to != address(this));
      require(_owns(msg.sender, _tokenId));
      _transfer(msg.sender, _to, _tokenId);
}

 function totalSupply() public view returns (uint) {
      return fishes.length - 1;
 }
```

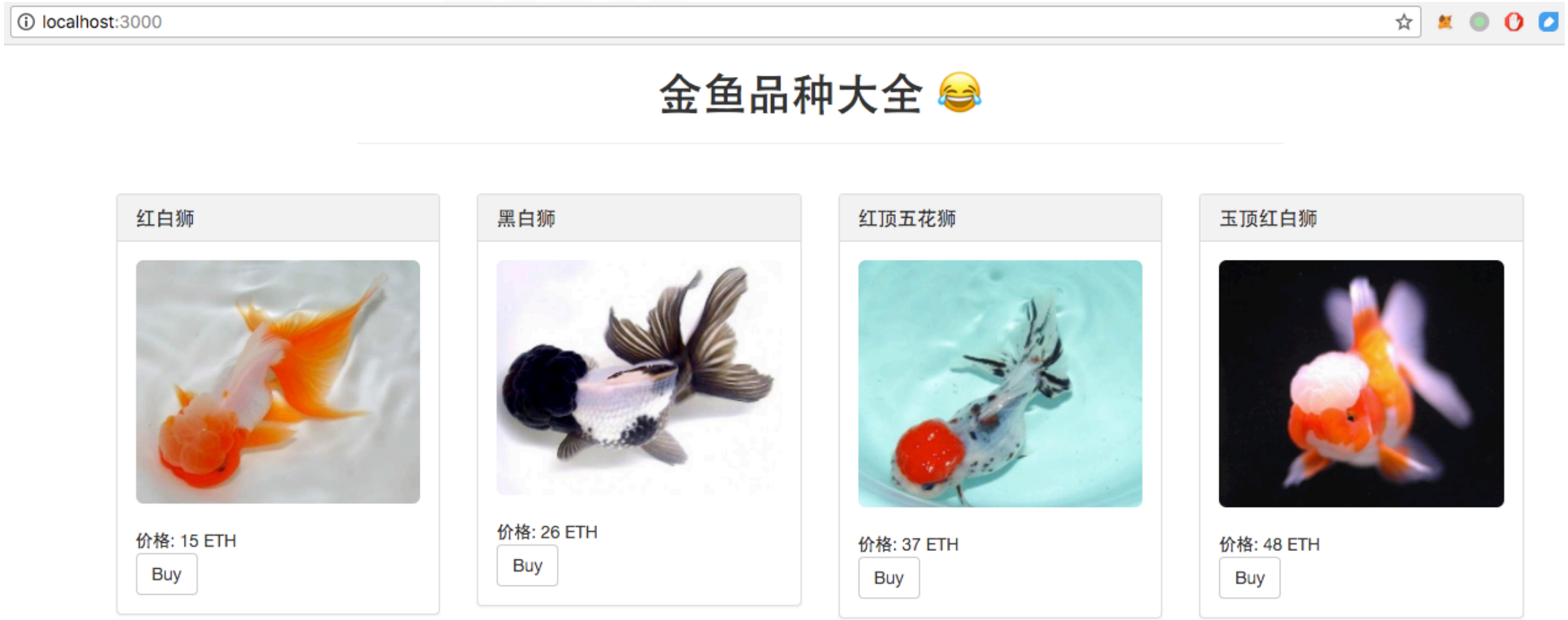
金鱼品种大全—FishCore-3

```
function tokensOfOwner(address _owner) external view returns(uint256[] ownerTokens) {
       uint256 tokenCount = balanceOf(_owner);
      if (tokenCount == 0) {
          return new uint256[](0);
      } else {
            uint256[] memory result = new uint256[](tokenCount);
            uint256 totalCats = totalSupply();
          uint256 resultIndex = 0;

          uint256 fishId;

          for (fishId = 1; fishId <= totalCats; fishId++) {
              if (fishIndexToOwner[fishId] == _owner) {
                  result[resultIndex] = fishId;
                  resultIndex++;
              }
          }
          return result;
      }
  }
```

金鱼品种大全—FishCore-4

```
function buyFish(uint256 _fishId) external payable returns (bool) {
        Fish storage fish = fishes[_fishId];
        require(msg.value == fish.price);
        address owner = fishIndexToOwner[_fishId];
        msg.sender.transfer(fish.price);
        _transfer(owner, msg.sender, _fishId);
}
```

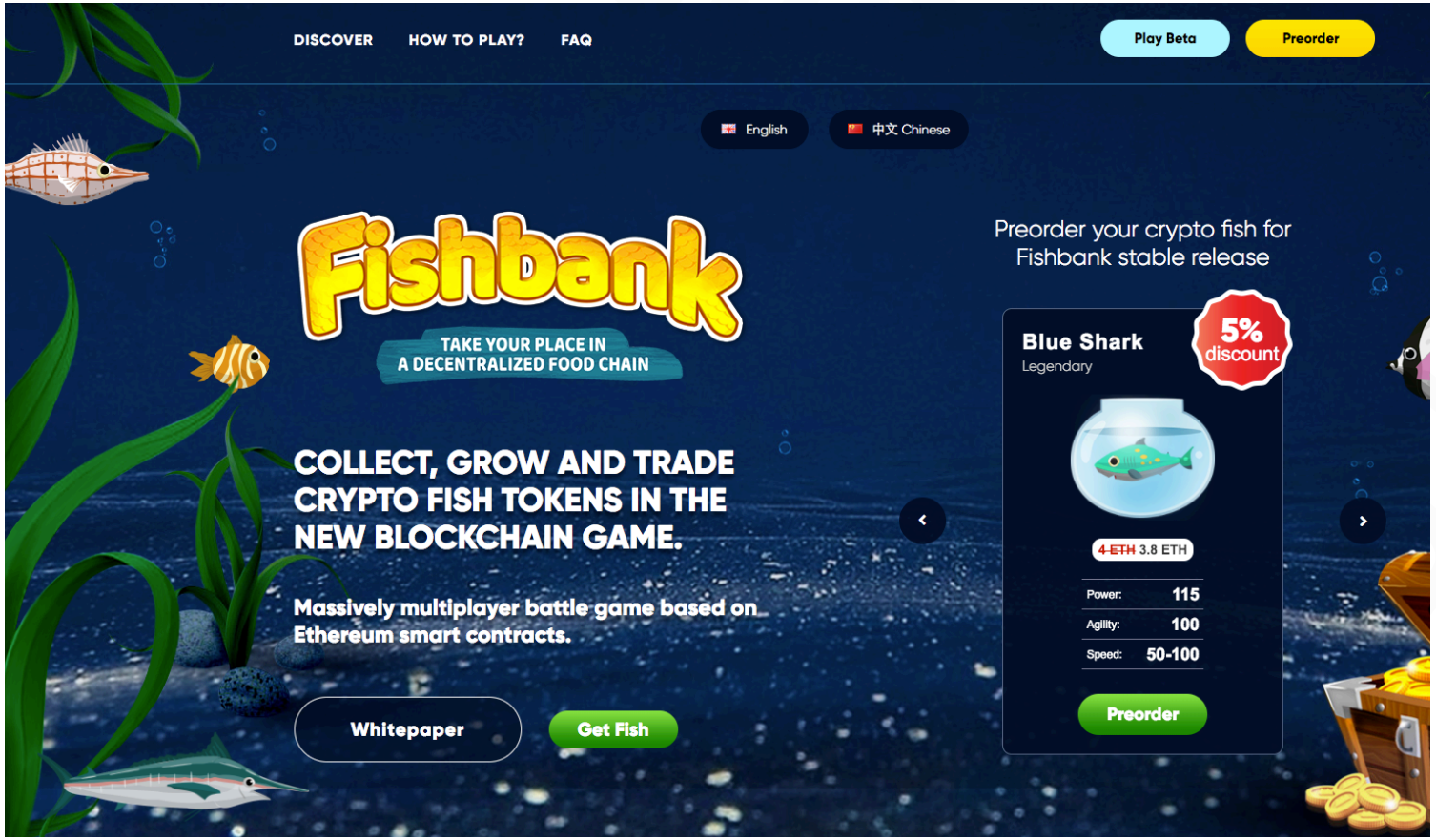
金鱼品种大全—UI


localhost:3000
金鱼品种大全 😂
红白狮
价格: 15 ETH
Buy
黑白狮
价格: 26 ETH
Buy
红顶五花狮
价格: 37 ETH
Buy
玉顶红白狮
价格: 48 ETH
Buy

下一个是谁？

招财猫？莱茨狗？

FishBank



2018年2月28日

智能合约源代码公布在GitHub

2018年3月18日

正式版部署到以太坊主网

Q&A