

NFT DAO

1.市场现状

2.NFT DAO简介

3.主要模块

share

proposal

vote

execute

4.提案执行流程

1.市场现状

当前的NFT市场上，有一部分NFT每一个都拥有极高的价值，比如当前元宇宙土地相关的项目，他们每一块土地的价格都及其高昂，因而对于普通玩家来说，即便看好元宇宙土地的前景，但是因为没有足够的资金实力，也没法进场。

而对于雄厚的资金拥有者来说，他们虽然有能力购买他们看好的NFT，但是也往往会因为高昂的NFT价格占据了他们太多的流动资金进而出现资金压力的情况。

而且对于由于NFT天生流动性不足且价格高昂的特性，很容易让用户错误估计NFT的实际价值，进而导致NFT价格的不合理

2.NFT DAO简介

根据NFT流动性不足的特点，我们的方案借鉴了之前很火的NFT碎片化的方法，将NFT转换成了实现了erc20协议的股份(share)。

由于持有股份意味着你对某个物品拥有部分的所有权，所以在我们的方案设计中，持有多少比例的股份就拥有多少的投票权重：

$$voteWeight = holdShares / totalShares$$

同时持有股份比例超过3%的人就可以通过propose函数创建一个提案(proposal)，该提案经过投票后，投票结果如果赞成票占总票数超过50%就会被执行，这也就意味着任何NFT持有者通过NFT DAO将NFT转换成shares之后，只要保留50%的票数就可以仍然保留对该NFT的控制权。

3.主要模块

整个项目主要分为4个模块 share 、 proposal 、 vote 、 execute 四个模块

share

share模块的主要功能是验证传入的合约地址是否是erc721合约，如果是则会调用NFT的转账函数，将该NFT转给新创建的share合约，同时会根据nft所有者的发行需求，设置可供mint的股份数量。当前支持两种share发行方式，第一种是nft所有者，将所有发行的股份数量都mint给自己，然后去uniswap中自己作为lp为当前交易对提供流动性，另一种方法是nft所有者只mint一部分share给自己，并给share用以太坊定价(这个价格是不变的)，其他想购买该share的用户将可以通过调用该mint函数支付对应的ether获取share.

proposal

proposal 模块主要功能是创建一个提案,可以通过以下函数进行创建

```
1  //targets : 是目标合约的地址
2  //value : 执行该操作要携带的ether数量
3  //calldatas : 函数签名和入参
4  //description : 对该提案的描述
5  function propose(
6      address[] memory targets,
7      uint256[] memory values,
8      bytes[] memory calldatas,
9      string memory description
10  )
11
```

CSS [复制代码](#)

要成功调用该函数需要持有至少3%的股份.

Proposal 一共有以下几种状态:

```
1  enum ProposalState {
2      Pending,
3      Active,
4      Canceled,
5      Defeated,
6      Succeeded,
7      Queued,
8      Expired,
9      Executed
10 }
```

Pending proposal已经创建投票还没有开始，这个时间当前设置为等待240个区块时间。

Active 可以进行投票；当前可以进行投票的时间段设置为11520个区块的时间

Canceled 取消投票，提案发起人可以调用

Defeated 投票结束但是提案没有被通过

Succeeded 投票结束且该提案已经被通过

Queued 该状态未启用

Executed 当proposal执行成功后会显示该状态

vote

vote 模块主要功能是对proposal进行投票，通过调用以下函数进行投票

```
1  // proposalId: 提案id
2  // support: 投票类型（赞成票 反对票 弃权票）
3
4  function castVote(uint256 proposalId, uint8 support) public virtual
   override returns (uint256) {
5      address voter = _msgSender();
6      return _castVote(proposalId, voter, support, "");
7  }
```

vote可以对所有在投票期的提案进行投票，投票时以当前账户最新的状态进行计票。

execute

execute模块主要功能是执行已经被通过的提案，可以通过以下函数进行调用

```

1  //targets   : 是目标合约的地址
2  //value    : 执行该操作要携带的ether数量
3  //calldatas : 函数签名和入参
4  //description : 对该提案的描述
5  function execute(
6      address[] memory targets,
7      uint256[] memory values,
8      bytes[] memory calldatas,
9      bytes32 descriptionHash
10 )

```

实际调用是通过call执行，如下：

```

1  targets[i].call{value: values[i]}(calldatas[i]);

```

需要注意只要在提案没被执行前，提案所有人就可以取消该提案。

4.提案执行流程

