

Homework 6 Nadine Chancay

Badger NFT

We now want to create an NFT.

You can use the Open Zeppelin libraries to help with this.

1. Create a new project in the IDE of your choice
2. Create a BadgerNFT contract this should inherit from any ERC721 implementation from the Open Zeppelin standard libraries
3. Give your NFT a name and a symbol.
4. Write unit tests to check that you can
5. Deploy your contract to the test network and send some NFTs to your colleagues.

For now we won't associate a digital asset with our token.

```
// contracts/GameItem.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract BadgerNFT is ERC721URIStorage {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;
    uint256 newItemId;
    address owner;

    struct NftInfo {
        address recipientAddress;
        uint256 itemId;
    }

    event TransferNotification(uint256, address);

    mapping(address => NftInfo[]) public nftInfo;
    mapping (address => uint256) items;

    constructor() ERC721("BadgerNFT", "BGN") {
        owner = msg.sender;
        newItemId=0;
    }
}
```

```

        items[msg.sender] = newItemId;
    }

    function awardItem(address user, string memory tokenURI)
        public
        returns (uint256)
    {
        newItemId = _tokenIds.current();
        _mint(user, newItemId);
        _setTokenURI(newItemId, tokenURI);
        _tokenIds.increment();
        return newItemId;
    }

    function transferItem(address userTo) public {
        items[userTo] += newItemId;
        items[msg.sender] = 0;
        nftInfo[msg.sender].push(NftInfo({recipientAddress: msg.sender, itemId: 0}));
        nftInfo[userTo].push(NftInfo({recipientAddress: userTo, itemId: newItemId}));
        emit TransferNotification(newItemId, userTo);
    }

    function getInfo(address _address) public view returns (NftInfo[] memory) {
        return nftInfo[_address];
    }
}

```