

# Blockchain HW1

資工四 110590004 林奕廷

\* All the contracts in this assignment are deployed on the Sepolia testnet. You can view them on Sepolia Etherscan by clicking the contract address.

\* All the contracts and websites can be accessed from the PepeCard repository <https://github.com/Error0229/PepeCard>

## Task 1: Deploy ERC20 Token

Contract Address: 0x8a899A996b233bFFc73cBDEf0BDB8817fcd6a4Fb

The screenshot displays the PepeCard web interface. At the top left is the 'PepeCard' logo. The main area features a 5x5 grid of cards, each with a question mark. The second card in the second row is highlighted with a checkmark. To the right of the grid is a 'Player Info' panel containing the account address, token balance (998.125 PC), buy-in (10,000 PC), safe flips (1), accumulated rewards (45.673 PC), and the next card reward (+137.019 PC). Below this is an 'Admin Controls' panel with buttons for 'Take Out ETH' and 'Mint Free Tokens'. At the bottom left is a 'Game Controls' panel with buttons for 'Leave Game & Collect Rewards', 'Redeem Tokens (PC)', and 'Redeem for ETH'. At the bottom right is a 'Purchase Tokens' panel with an input field for 'ETH Amount', a 'Purchase Tokens' button, and a rate of 0.0001 ETH = 1 PC.

Demo website: <https://pepe-card.vercel.app/>

## Game Description

PepeCard is a risk-reward card flipping game implemented as a smart contract. Here's how it works:

### Game Rules

- Players start by buying in with tokens (minimum 100 wei)
- There's a 5x5 grid of cards (25 total)
- One card is randomly selected as the "ghost card"
- Players flip cards one at a time

- Each safe flip earns increasing rewards
- If the ghost card is flipped, the game ends and all accumulated rewards are lost
- Players can “Leave” at any time to collect their accumulated rewards

### Reward Mechanism

- Rewards increase quadratically with each successful flip
- House edge is 5%
- Base reward formula:  $A * (n^2 + n)$ , where:
  - $A = (25 * \text{buyin} * 0.95) / (5200 * 2)$
  - $n$  = number of successful flips
- Players can redeem(burn) tokens for ETH at a rate of 10000:1 with a 2% redemption fee


### Only Owner Functions

- TakeOut(): Allows the owner to withdraw all ETHs from the contract
- FreeToken(): Mints 10000 tokens to the owner

### Failed Transaction

When a non-owner tries to call the FreeToken() function, the transaction will fail with OwnableUnauthorizedAccount message. ex:

0xee9389db90693ab65c2addf7b9cd97aabe6e1836413a574924c8b5524aa75829


Home

Transaction Details
<
>

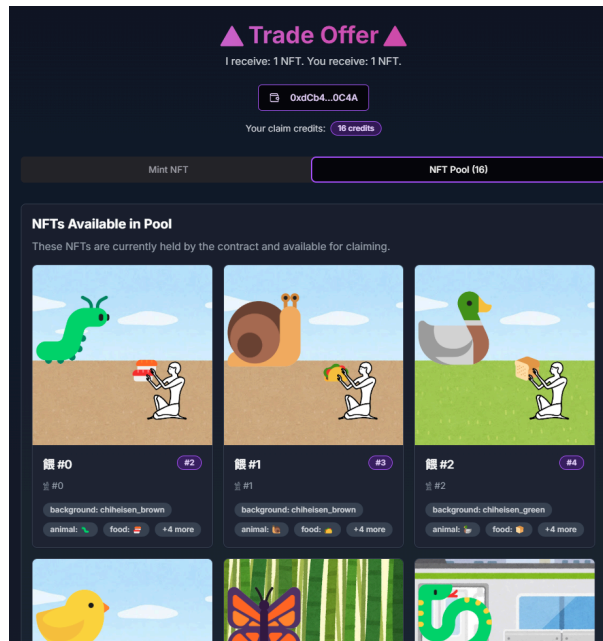
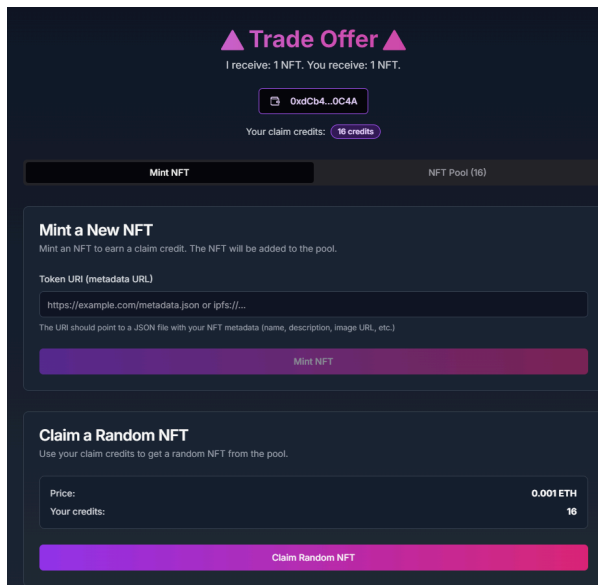
Overview
State

[ This is a Sepolia Testnet transaction only ]

Transaction Hash:	0xee9389db90693ab65c2addf7b9cd97aabe6e1836413a574924c8b5524aa75829
Status:	Fail with Custom Error 'OwnableUnauthorizedAccount (account=0x6C82a1A792388880E1BD1B7725bdC234E0601325)'
Block:	7935317 77364 Block Confirmations
Timestamp:	10 days ago (Mar-19-2025 01:07:00 PM UTC)
Transaction Action:	Call Free Token Function by 0x6C82a1A7...4E0601325 on 0x8a899A99...7fcd6a4Fb
From:	0x6C82a1A792388880E1BD1B7725bdC234E0601325
To:	0x8a899A996b233bFFc73cBDEf0BDB8817fcd6a4Fb
	Warning! Error encountered during contract execution [execution reverted]
Value:	0 ETH
Transaction Fee:	0.000044309407381512 ETH
Gas Price:	1.860489057 Gwei (0.000000001860489057 ETH)

## Task 2: Deploy ERC721 Token

Contract Address: 0xdb319C816aec7cb4F44bC8e825Fe00248457E837



Demo website: <https://trade-offer-ten.vercel.app/>

The Trade Offer contract implements a unique NFT trading system where users can mint and Gacha NFTs.

### Core Mechanics

- Users can mint new NFTs by providing metadata URIs
- Each minted NFT is held by the contract and adds to the pool
- Minters receive 1 claim credit for each NFT they mint
- Users can claim random NFTs from the pool using their credits (additional fee needed)

### Only Owner Functions

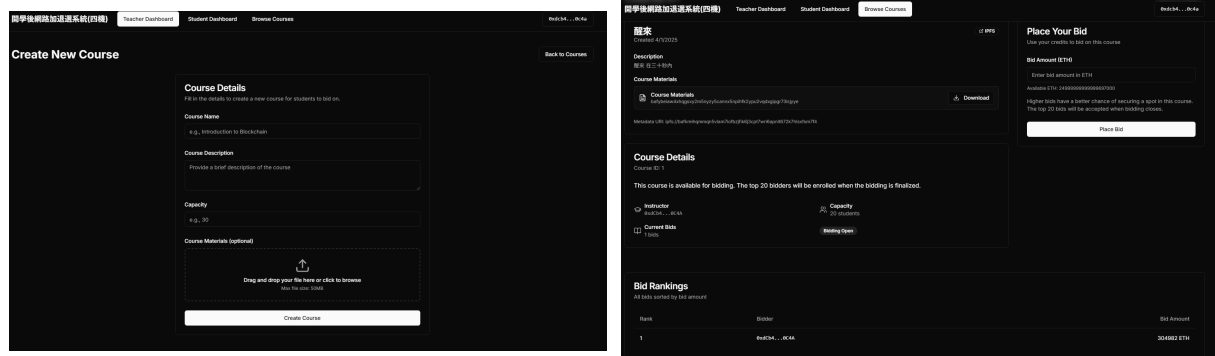
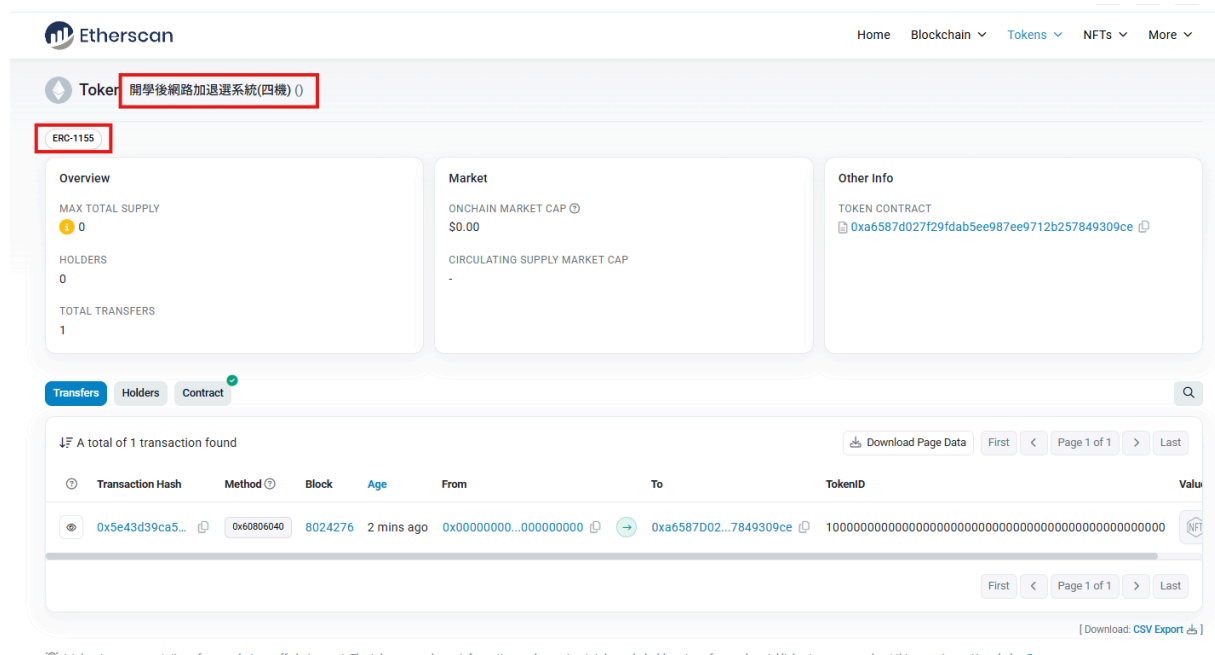
- `withdraw()`: Allows the owner to withdraw all ETHs from the contract

### Known Issues

- Since you getting a different randomness from the contract every time interacting with it, you may repeatedly try to claim an NFT until you get the one you want.

\* Note: You may check the NFTs on the OpenSea testnet by clicking [this text](#).

Contract Address: 0xa6587d027f29fdab5ee987ee9712b257849309ce



Demo website: <https://isms-nagios.vercel.app/>

The 開學後網路加退選系統(四機) implements a course bidding system with the following features:

## Core Mechanics

- Students must register to receive 25 ETH initial credits
- Teachers can create courses with specified capacity and metadata
- Students bid on courses using their credits
- Credits are locked during active bids
- When bidding ends, top N bidders (based on capacity) win course slots
- Losing bidders get their credits refunded

## Key Functions

## 1. Registration

- New students receive 25 ETH initial credits
- Each address can only register once

## 2. Course Creation

- Only addresses with TEACHER\_ROLE can create courses
- Teachers specify capacity and course metadata
- Each course gets a unique courseId
- NFTs are minted to represent course slots

### 3. **Bidding Process**

- Students can bid multiple times on the same course
- Subsequent bids are added to their existing bid amount
- Credits are locked during active bids
- Must have sufficient credits to place bid

### 4. **Finalization**

- Only course teacher can finalize bidding
- System automatically selects top N highest bidders
- Winners receive course NFT as proof of enrollment
- Losing bidders get credits refunded
- Course status changes to “finalized”

## Task 4: Deploy ERC721A Token

Contract Address: 0x6a3E113c919E03b3270359461DcacBaaf31C8469

No Demo website available.

- The contract is a simple ERC721A implementation with maximum supply of 500 NFTs.
- Only function implementation is `airdrop(to, quantity)` which allows the owner to mint NFTs to a specified address.

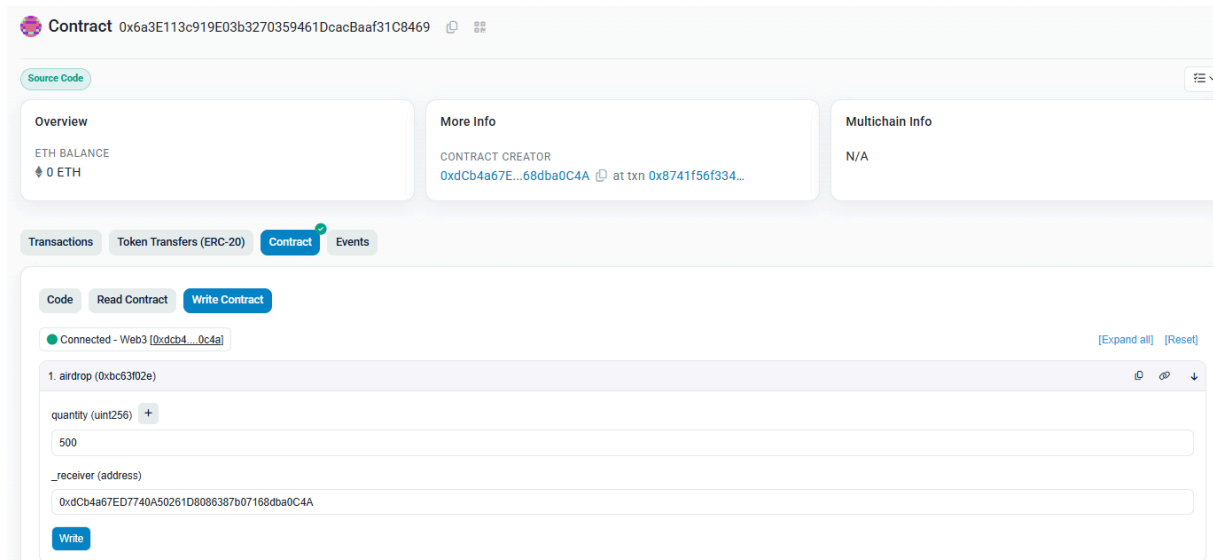


Figure 1: The contract

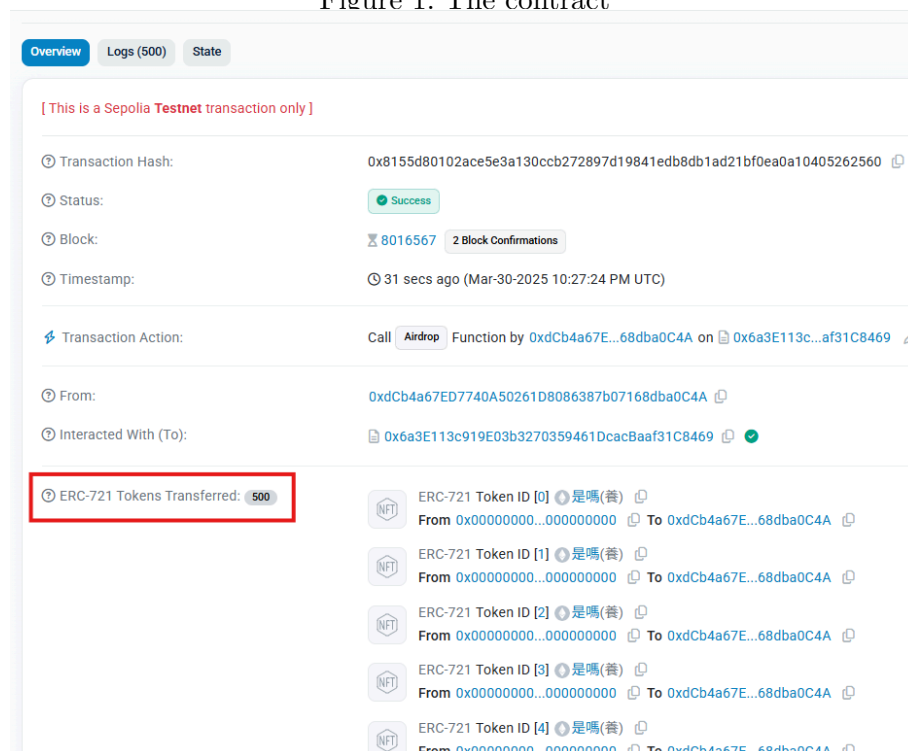


Figure 2: 500 NFTs minted in single transaction

You can check the batch minted NFTs on the OpenSea testnet by the following link: <https://testnets.opensea.io/ja/collection/shi-ma>