Cryptozombies

Chapter 2: Zombies attack their victims

Contents

Addresses

Each contract has an address on the chain. It's unique for it. The tutorial doesn't explain the meaning yet but here's cryptozombies team's account address

```
0x0cE446255506E92DF41614C46F1d6df9Cc969183
```

In the first chapter we learn what a mapping is (basically a dictionary). It holds mappings between type1 and type2. We're going to need 2 mappings. The first one holds the zombie id to owner address tuples. The second one memorizes how many zombies each owner has.

```
    mapping (uint => address) public zombieToOwner;
    mapping (address => uint) ownerZombieCount;
```

msg.sender

During a function call, we can get the caller's address (person or contract). It's in a variable called sender that's part of msg.

require

require simply validates that the inner condition is fulfilled. If it is not, the function execution is canceled.

String comparison

Oh also, for some reason, there's no string comparison in solidity. We have to compare the keccak256 hashes of 2 strings in order to actually compare them.

Inheritance

A contract can inherit from another contract. There are also 4 types of visibility.

Visibility

- Public
  - The function is publicly callable and also is inherited by every subclass
- External
  - The function is callable only from outside the contract.
- Internal
  - The function is internal callable, so only callable inside the contract or other contracts that inherit from it
- Private
  - The function is only callable from inside the current contract

Data location

Storage refers to variables stored permanently on the blockchain. Memory variables are temporary, and are erased between external function calls to your contract.

Interfaces

To call functions from other contracts, we must define an interface for it. Well, much like an interface from Java, we only need to define the signatures and end the lines with a semicolon, no function bodies.

Multiple return values

Functions in solidity can return multiple variables. We simply need to define in the signature which variables we will return, with commas in between. Here's an example from the tutorial:

```
function multipleReturns() internal returns(uint a, uint b, uint c) {
  return (1, 2, 3);
}
```

If we only needed the second value from the returned 'list', we need to catch it like this (,secondValue,).

CryptoKitties

I'm still in shock after seeing some of the prices on their website. A pure-breed cat is way cheaper. But I get the point of NFT.

Anyway, we're using their contract in our contract, because our zombies love feline meat.