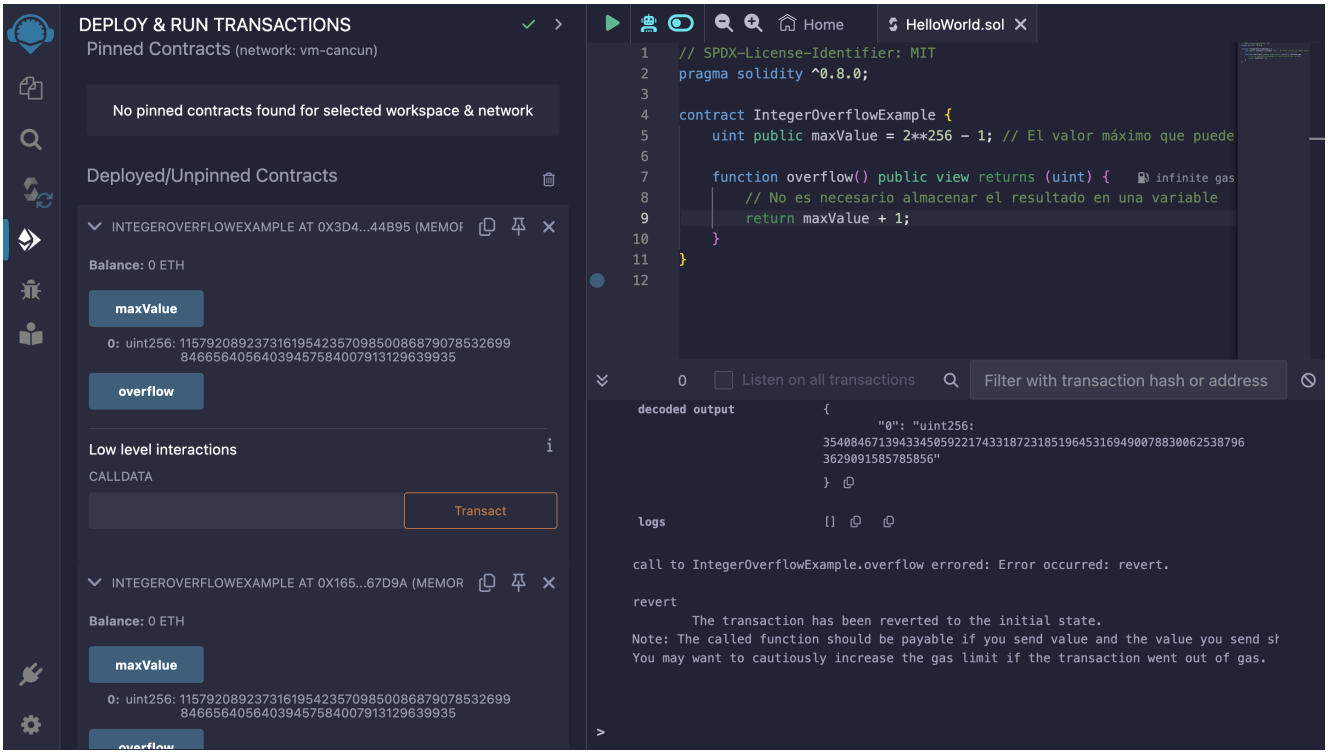
Integer Overflow en Solidity

[crayon-6a9d4ed522c40937910450/]

Este contrato de Solidity ilustra un ejemplo de desbordamiento de enteros. Veamos el código paso a paso:

1. **contract IntegerOverflowExample:** Aquí se define el contrato IntegerOverflowExample.
2. **uint public maxValue =** 2**256 – 1 En esta línea se declara una variable pública maxValue de tipo uint (entero sin signo de 256 bits) y se le asigna el valor máximo que puede contener un uint256. Este valor se calcula como 2 elevado a la potencia de 256 menos 1. Es importante destacar que el valor máximo es extremadamente grande y prácticamente imposible de representar en la práctica.
3. **function overflow() public view returns (uint):** Aquí se define una función pública llamada overflow, que no modifica el estado del contrato (view) y devuelve un valor de tipo uint.
4. **return maxValue + 1;** Dentro de la función overflow, se intenta realizar una operación de suma con maxValue y 1. Dado que maxValue ya es el valor máximo que puede contener un uint256, sumarle 1 resultará en un desbordamiento de enteros.


DEPLOY & RUN TRANSACTIONS
Pinned Contracts (network: vm-cancun)
No pinned contracts found for selected workspace & network
Deployed/Unpinned Contracts
INTEGEROVERFLOWEXAMPLE AT 0X3D4...44B95 (MEMOF
Balance: 0 ETH
maxValue
0: uint256: 115792089237316195423570985008687907853269984665640564039457584007913129639935
overflow
Low level interactions
CALLDATA
Transact
INTEGEROVERFLOWEXAMPLE AT 0X165...67D9A (MEMOR
Balance: 0 ETH
maxValue
0: uint256: 115792089237316195423570985008687907853269984665640564039457584007913129639935
Home
HelloWorld.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract IntegerOverflowExample {
uint public maxValue = 2**256 - 1; // El valor máximo que puede
function overflow() public view returns (uint) {
infinite gas
// No es necesario almacenar el resultado en una variable
return maxValue + 1;
}
}
0
Listen on all transactions
Filter with transaction hash or address
decoded output
{
"0": "uint256:
3540846713943345059221743318723185196453169490078830062538796
3629091585785856"
}
logs
[]
call to IntegerOverflowExample.overflow errored: Error occurred: revert.
revert
The transaction has been reverted to the initial state.
Note: The called function should be payable if you send value and the value you send sh
You may want to cautiously increase the gas limit if the transaction went out of gas.
>