Reducing gas waste in smart contracts

11th, November 2022 **Reported By: Syed Ali Hamdani**

Two smart contracts have been made and tested on https://remix.ethereum.org Which is an internet-based ide. Both contracts have been deployed by Ali on 11th, November 2022. Both contracts were making the same token. The unoptimizedCoin contract code was written without using the optimized code and the OptimizedCoin contract was written with optimized code. The purpose was to show how many gas units could be saved with optimized code.

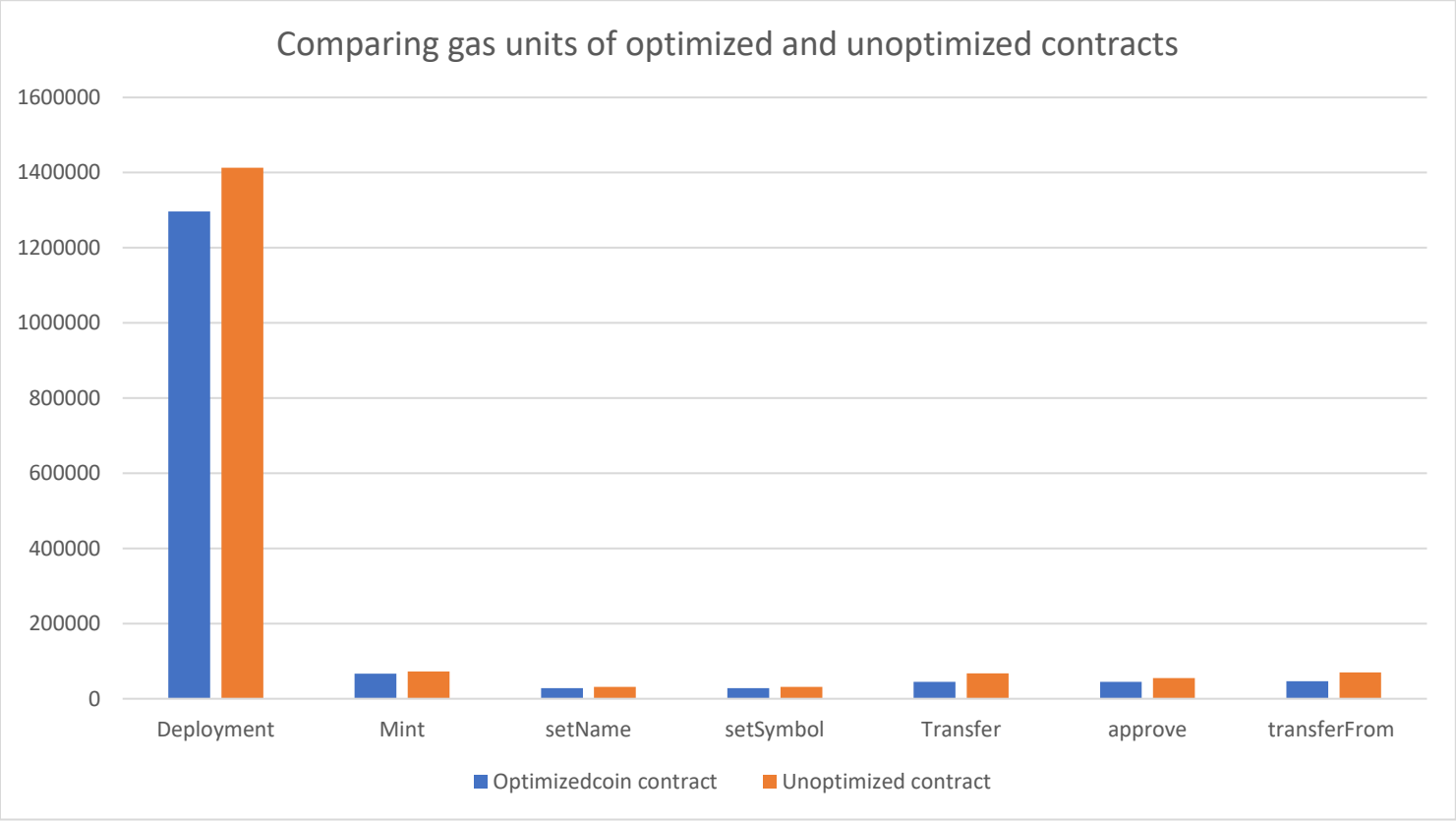
3 rules have been used to the contract optimized and saved some gas units.

1. **Space-for-Time Rule**
2. **Logic Rule 5: Boolean variable Elimination**
3. **Expression Rule 3: [context-dependent Rule]**

Space-for-Time rule: This rule saved gas units by managing the space in the OptimizedCoin smart contract. In the blockchain, storage space is more expensive than memory space. By using this rule, less storage space was used as It could be to reduce the gas units.

Logic Rule 5: Boolean variable Elimination: By using this rule all the Boolean variables were removed. Instead of a Boolean variable direct evaluation of a condition was used. This reduces the code size of the smart contract.

Expression Rule 3: [context-dependent Rule]: Its basic idea is to avoid the repeated calculation of the same expression by saving the result of the first invocation and then reusing this result later on. Avoiding repeated calculations helps to reduce the size of the bytecode of the smart contract. If the bytecode size is reduced the gas units also reduce by themselves.



Name of actions	Unoptimizedcoin contract (gas units)	Optimizedcoin contract (gas units)	% Gas saved
Deployment	1413172	1296981	8.22
Mint(function call)	72639	66833	7.99
setName(function call)	31719	28148	11.26
setSymbol(function call)	31685	28114	11.27
Transfer(function call)	67672	45289	33.08
approve(function call)	55136	44878	18.60
transferFrom (function call)	70171	46759	33.36

Most of the gas is used at the time of contract deployment. If the contract is optimized it can save up to 8% gas. There are two types of functions in the smart contract read functions and write functions. Read functions only read the data from the blockchain and do not consume any gas. That is why I did not mention those functions' names here. Write functions, and write the data inside the blockchain that uses the gas as an execution cost. There are 6 write functions in both smart contracts. All written function names are mentioned above as gas they use during the function call.

These rules can save 8 to 33 percent of the gas in contracts. 8% may not bigger amount in this simple smart contract but in the bigger contracts, it will be very big. By using these rules, we can save many gas units in bigger contracts and complete our tasks on time.