Overview

This document provides a detailed outline of the required update requirements across 10 sample contracts. Developers are expected to carefully implement each modification to ensure these contracts operate correctly maintenance.

Among these, the state updates cover various aspects, including the addition, removal, or modification of specific variables. These changes span adjustments in data types, visibility modifiers, names, and initial values.
This document categorizes the main types of state updates as follows:

1. Add Variable: new variables are introduced in certain contracts;
2. Remove Variable: delete the existing state variable;
3. Update Variable: modifying the existing state into new version;

Requirements

1- Hello

Revised Code:

```
contract Hello {
    string public text;
    uint256 public num;
    function initialize() public {
        text = "Hello";
        num = 123;
    }
    function doSomething() public {
        num += 1;
    }
    function getSender() public view returns (address) {
        return msg.sender;
    }
}
```

2-Wallet

Revised Code:

```
contract Wallet {
    address owner;
    mapping(address => uint8) balances;
```

```
    constructor() public {
        owner = msg.sender;
    }
    event Deposited(address indexed user, uint256 amount);
    event Withdrawn(address indexed user, uint256 amount);
    event Migrated(address indexed to, uint256 amount);
    function deposit() public payable {
        require(msg.value > 0, "Deposit must be greater than 0");
        balances[msg.sender] += msg.value;
        emit Deposited(msg.sender, msg.value);
    }
    function withdraw(uint256 amount) public {
        require(amount <= balances[msg.sender], "Insufficient balance");
        balances[msg.sender] -= amount;
        payable(msg.sender).transfer(amount);
        emit Withdrawn(msg.sender, amount);
    }
    function getBalance(address user) public view returns (uint256) {
        return balances[user];
    }
}
```

3-Todos

Revised Code:

```
contract Todos {
    struct Todo {
        string text;
        bool completed;
    }
    Todo[3] public todos;

    function get(uint256 _index) public view returns (string memory text, bool completed){
      require(_index < todos.length, "Index out of bounds");
      Todo storage todo = todos[_index];
      return (todo.text, todo.completed);
    }
    function create(string calldata _text) public {
        todos.push(Todo({text: _text, completed: false}));
    }
```

```
    function updateText(uint256 _index, string calldata _text) public {
        Todo storage todo = todos[_index];
        todo.text = _text;
    }

    function toggleCompleted(uint256 _index) public {
        Todo storage todo = todos[_index];
        todo.completed = !todo.completed;
    }
}
```

4-BancorBuyer

Add Variables

- Data Type：uint
- Visibility Modifier：public
- Variable Name：reward
- Value：null

Revised Code:

```
function add_reward() payable {
    reward += msg.value;
}
```

5-Grid

Remove Variables:

- Data Type：struct
- Visibility Modifier：none
- Variable Name：User
- Value：null

Update Variables:

Old version：

- Data Type：mapping(address => User)
- Visibility Modifier：none
- Variable Name：users
- Value：null

New version：

- Data Type： mapping(address => uint)
- Visibility Modifier： none
- Variable Name： pendingWithdrawals
- Value： null

Revised Code:

```
function checkPendingWithdrawal() constant returns (uint) {
      return pendingWithdrawals[msg.sender];
}
function withdraw() {
      if (pendingWithdrawals[msg.sender] > 0) {
         uint amount = pendingWithdrawals[msg.sender];
         pendingWithdrawals[msg.sender] = 0;
         msg.sender.transfer(amount);
      }
 }
```

6-MyartPoint

Remove Variables:

1、- Data Type： uint
- Visibility Modifier： none
- Variable Name： number
- Value： 0

2、- Data Type： mapping(uint => address)
- Visibility Modifier： private
- Variable Name： indices
- Value： null

3、- Data Type： mapping(address => bool)
- Visibility Modifier： private
- Variable Name： exists
- Value： null

Revised Code:

remove functions: getAdrByIndex; recordNewAddress

modified functions:

```
function allocate(address to, uint amount) public onlyOwner {
        require(to != address(0));
        require(!frozenAccount[to]);
        require(!halted && amount > 0);
        require(balances[owner] >= amount);
        balances[owner] = balances[owner].sub(amount);
        balances[to] = balances[to].add(amount);
        emit Transfer(address(0), to, amount);
 }
```

7-BancorKillerContract

Remove Variables:

1、- Data Type：address
- Visibility Modifier：public
- Variable Name：base_token
- Value：null

2、- Data Type：mapping(address => uint256)
- Visibility Modifier：pubic
- Variable Name：token_balance
- Value：null

Add Variables:

3、- Data Type：uint256
- Visibility Modifier：none
- Variable Name：eth_balance
- Value：null

4、- Data Type：uint256
- Visibility Modifier：none
- Variable Name：traded_token_balance
- Value：null

Revised Code:

remove functions: market_is_open; seed_base_token

modified functions:

```
function get_amount_sell(uint256 _amount) public view returns(uint256) {
   uint256 eth_balance_ = eth_balance;
   uint256 traded_token_balance_ = traded_token_balance;
   uint256 traded_token_balance_plus_amount_=traded_token_balance_ + _amount;
   return    (2*eth_balance_*_amount)/(traded_token_balance_+
               traded_token_balance_plus_amount_);
 }
function get_amount_buy(uint256 _amount) public view returns(uint256) {
   uint256 eth_balance_ = eth_balance;
   uint256 traded_token_balance_ = traded_token_balance;
   uint256 eth_balance_plus_amount_ = eth_balance_ + _amount;
   return (_amount*traded_token_balance_*(eth_balance_plus_amount_ +
          eth_balance_))/(2*eth_balance_plus_amount_*eth_balance_);
}
```

8-Tiles

Update Variables:

Old version:

- Data Type：Tile[16][16]
- Visibility Modifier：public
- Variable Name：tiles
- Value：null

New version:

- Data Type：Tile[8][8]
- Visibility Modifier：public
- Variable Name：tiles
- Value：null

9-Factory

Update Variables:

Old version：

- Data Type：address
- Visibility Modifier：private
- Variable Name：owner1
- Value：0x6CAa636cFFbCbb2043A3322c04dE3f26b1fa6555

Old version :

- Data Type：address
- Visibility Modifier：private
- Variable Name：owner2
- Value：0xbc2d90C2D3A87ba3fC8B23aA951A9936A6D68121

Old version :

- Data Type：address
- Visibility Modifier：private
- Variable Name：owner3
- Value：0x680d821fFE703762E7755c52C2a5E8556519EEDc

New version :

- Data Type：address[]
- Visibility Modifier：public
- Variable Name：owners
- Value：[0x6CAa636cFFbCbb2043A3322c04dE3f26b1fa6555,
  0xbc2d90C2D3A87ba3fC8B23aA951A9936A6D68121,
  0x680d821fFE703762E7755c52C2a5E8556519EEDc]

Revised Code:

```
modifier onlyOwnerOrCreator {
    require(msg.sender == owners[0] || msg.sender == owners[1] || msg.sender ==.
            owners[2] || msg.sender == creator);
    _;
}
```

10-CryptoElections

Add Variables:

1、- Data Type：bool
- Visibility Modifier：none
- Variable Name：transferEnabled
- Value：false

2、- Data Type：bool
- Visibility Modifier：none
- Variable Name：inited

- Value：false

Update Variables:

Adding in Struct City:

- Data Type：uint
- Visibility Modifier：none
- Variable Name：startPrice
- Value：null

- Data Type：uint
- Visibility Modifier：none
- Variable Name：multiplierStep
- Value：null

Revised Code:

```
function allowance(address tokenOwner, address spender)
   public constant returns (uint remaining) {
            require(transferEnabled);
            return allowed[tokenOwner][spender];
}
function addOldMayors(uint[] citiesIds,uint[] purchases,address[] mayors) public
      onlyCreator()
{
           require(!inited);
           for (uint i = 0;i<citiesIds.length;i++) {
                cities[citiesIds[i]].mayor = mayors[i];
                cities[citiesIds[i]].purchases = purchases[i];
           }
 }
```