```
1 // SPDX-License-Identifier: AGPL-3.0-or-later
                                                            1 // SPDX-License-Identifier: AGPL-3.0-or-later
3 pragma solidity 0.7.5;
                                                             3 pragma solidity 0.7.5;
 5 library LowGasSafeMath {
                                                             5 library LowGasSafeMath {
    /// @notice Returns x + y, reverts if sum overf
                                                                /// @notice Returns x + y, reverts if sum overf
   lows uint256
                                                               lows uint256
     /// @param x The augend
                                                                   /// @param x The augend
8
      /// @param y The addend
                                                             8
                                                                   /// @param y The addend
      /// @return z The sum of x and y
                                                             9
                                                                   /// @return z The sum of x and y
      function add(uint256 x, uint256 y) internal pur
                                                                   function add(uint256 x, uint256 y) internal pur
   e returns (uint256 z) {
                                                               e returns (uint256 z) {
          require((z = x + y) >= x);
                                                            11
                                                                       require((z = x + y) >= x);
13
                                                            13
     function add32(uint32 x, uint32 y) internal pur
                                                                 function add32(uint32 x, uint32 y) internal pur
14
                                                            14
   e returns (uint32 z) {
                                                               e returns (uint32 z) {
          require((z = x + y) >= x);
                                                            15
                                                                       require((z = x + y) >= x);
15
16
                                                            16
17
                                                            17
       /// @notice Returns x - y, reverts if underflow
                                                            18
                                                                   /// @notice Returns x - y, reverts if underflow
      /// @param x The minuend
                                                                   /// @param x The minuend
19
                                                            19
20
       /// @param y The subtrahend
                                                            20
                                                                   /// @param y The subtrahend
21
       /// @return z The difference of x and y
                                                            21
                                                                   /// @return z The difference of x and y
       function sub(uint256 x, uint256 y) internal pur
                                                                   function sub(uint256 x, uint256 y) internal pur
   e returns (uint256 z) {
                                                               e returns (uint256 z) {
23
           require((z = x - y) <= x);
                                                            23
                                                                       require((z = x - y) <= x);
24
                                                            24
25
                                                            25
       function sub32(uint32 x, uint32 y) internal pur
                                                            26
                                                                   function sub32(uint32 x, uint32 y) internal pur
   e returns (uint32 z) {
                                                               e returns (uint32 z) {
27
         require((z = x - y) \le x);
                                                            27
                                                                       require((z = x - y) <= x);
28
                                                            28
29
                                                            29
30
      /// @notice Returns x * y, reverts if overflows
                                                            30
                                                                   /// @notice Returns x * y, reverts if overflows
      /// @param x The multiplicand
                                                                   /// @param x The multiplicand
31
                                                            31
32
      /// @param y The multiplier
                                                            32
                                                                   /// @param y The multiplier
33
      /// @return z The product of x and y
                                                            33
                                                                   /// @return z The product of x and y
       function mul(uint256 x, uint256 y) internal pur
                                                                   function mul(uint256 x, uint256 y) internal pur
   e returns (uint256 z) {
                                                               e returns (uint256 z) {
           require(x == 0 || (z = x * y) / x == y);
                                                                       require(x == 0 \mid | (z = x * y) / x == y);
35
                                                            35
36
37
                                                            37
       /// @notice Returns x + y, reverts if overflows
                                                                   /// @notice Returns x + y, reverts if overflows
38
                                                            38
   or underflows
                                                               or underflows
      /// @param x The augend
                                                                  /// @param x The augend
39
                                                            39
40
       /// @param y The addend
                                                            40
                                                                   /// @param y The addend
      /// @return z The sum of x and y
                                                                   /// @return z The sum of x and y
      function add(int256 x, int256 y) internal pure
                                                                   function add(int256 x, int256 y) internal pure
    returns (int256 z) {
                                                                returns (int256 z) {
           require((z = x + y) >= x == (y >= 0));
                                                                       require((z = x + y) >= x == (y >= 0));
43
                                                            43
44
                                                            44
46
       /// @notice Returns x - y, reverts if overflows
                                                            46
                                                                   /// @notice Returns x - y, reverts if overflows
   or underflows
                                                               or underflows
47
      /// @param x The minuend
                                                            47
                                                                  /// @param x The minuend
       /// @param y The subtrahend
                                                                   /// @param y The subtrahend
                                                            48
      /// @return z The difference of x and y
                                                                   /// @return z The difference of x and y
                                                            49
      function sub(int256 x, int256 y) internal pure
                                                                   function sub(int256 x, int256 y) internal pure
    returns (int256 z) {
                                                                returns (int256 z) {
          require((z = x - y) <= x == (y >= 0));
51
                                                            51
                                                                       require((z = x - y) <= x == (y >= 0));
52
                                                            52
53
                                                            53
```

```
e returns(uint256 z){
                                                                 e returns(uint256 z){
                                                                         require(y > 0);
55
           require(y > 0);
                                                              55
56
            z=x/y;
                                                              56
                                                                         z=x/y;
57
                                                              57
58 }
                                                              58 }
59
                                                              59
60 interface IERC20 {
                                                              60 interface IERC20 {
61
                                                              61
        function totalSupply() external view returns (u
                                                                     function totalSupply() external view returns (u
62
                                                              62
    int256):
                                                                 int256):
63
                                                              63
        function balanceOf(address account) external vi
                                                                      function balanceOf(address account) external vi
64
                                                              64
    ew returns (uint256);
                                                                  ew returns (uint256);
65
                                                              65
        function transfer(address recipient, uint256 am
                                                                     function transfer(address recipient, uint256 am
66
                                                              66
    ount) external returns (bool);
                                                                 ount) external returns (bool);
67
                                                              67
        function allowance(address owner, address spend
                                                                     function allowance(address owner, address spend
68
                                                              68
    er) external view returns (uint256);
                                                                  er) external view returns (uint256);
69
                                                              69
        function approve(address spender, uint256 amoun
                                                                      function approve(address spender, uint256 amoun
70
                                                              70
    t) external returns (bool);
                                                                 t) external returns (bool);
71
                                                              71
        function transferFrom(address sender, address r
                                                                     function transferFrom(address sender, address r
72
                                                              72
    ecipient, uint256 amount) external returns (bool):
                                                                 ecipient, uint256 amount) external returns (bool):
74
        event Transfer(address indexed from, address in
                                                              74
                                                                      event Transfer(address indexed from, address in
    dexed to, uint256 value);
                                                                 dexed to, uint256 value);
75
                                                              75
        event Approval(address indexed owner, address i
                                                                     event Approval(address indexed owner, address i
    ndexed spender, uint256 value);
                                                                 ndexed spender, uint256 value);
77 }
                                                              77 }
78
79 library Address {
                                                              79 library Address {
                                                              80
        function isContract(address account) internal v
                                                                     function isContract(address account) internal v
81
    iew returns (bool) {
                                                                  iew returns (bool) {
           // This method relies in extcodesize, which
                                                                         // This method relies in extcodesize, which
    returns 0 for contracts in
                                                                 returns 0 for contracts in
83
                                                              83
            // construction, since the code is only sto
                                                                         // construction, since the code is only sto
    red at the end of the
                                                                 red at the end of the
84
           // constructor execution.
                                                              84
                                                                         // constructor execution.
85
                                                              85
            uint256 size;
                                                                         uint256 size;
86
                                                              86
            // solhint-disable-next-line no-inline-asse
                                                                          // solhint-disable-next-line no-inline-asse
    mblv
                                                                 mblv
88
            assembly { size := extcodesize(account) }
                                                                         assembly { size := extcodesize(account) }
                                                              88
            return size > 0:
89
                                                              89
                                                                          return size > 0:
90
                                                              90
91
                                                              91
        function sendValue(address payable recipient, u
                                                                     function sendValue(address payable recipient, u
    int256 amount) internal {
                                                                  int256 amount) internal {
            require(address(this).balance >= amount, "A
                                                                         require(address(this).balance >= amount, "A
    ddress: insufficient balance");
                                                                 ddress: insufficient balance");
94
                                                              94
            // solhint-disable-next-line avoid-low-leve
                                                                         // solhint-disable-next-line avoid-low-leve
95
    l-calls, avoid-call-value
                                                                  l-calls, avoid-call-value
            (bool success, ) = recipient.call{ value: a
                                                                          (bool success, ) = recipient.call{ value: a
    mount }("");
                                                                 mount }("");
            require(success, "Address: unable to send v
                                                                         require(success, "Address: unable to send v
    alue, recipient may have reverted");
                                                                 alue, recipient may have reverted");
98
                                                              98
99
                                                              99
        function functionCall(address target, bytes mem
                                                             100
                                                                     function functionCall(address target, bytes mem
    ory data) internal returns (bytes memory) {
                                                                  ory data) internal returns (bytes memory) {
          return functionCall(target, data, "Address: l
                                                                       return functionCall(target, data, "Address: l
    ow-level call failed");
                                                                 ow-level call failed");
102
                                                             102
        }
                                                                     }
103
                                                             103
104
        function functionCall(
                                                             104
                                                                     function functionCall(
```

function div(uint256 x, uint256 y) internal pur

function div(uint256 x, uint256 y) internal pur

```
105
              address target,
                                                               105
                                                                            address target,
 106
                                                               106
              bytes memory data,
                                                                            bytes memory data,
 107
              string memory errorMessage
                                                               107
                                                                            string memory errorMessage
 108
          ) internal returns (bytes memory) {
                                                                        ) internal returns (bytes memory) {
 109
              return _functionCallWithValue(target, data,
                                                               109
                                                                            return _functionCallWithValue(target, data,
      0, errorMessage);
                                                                    0, errorMessage);
 110
 111
                                                               111
 112
          function functionCallWithValue(address target,
                                                               112
                                                                        function functionCallWithValue(address target,
       bytes memory data, uint256 value) internal returns
                                                                     bytes memory data, uint256 value) internal returns
      (bytes memory) {
                                                                    (bytes memory) {
 113
              return functionCallWithValue(target, data,
                                                               113
                                                                            return functionCallWithValue(target, data,
       value, "Address: low-level call with value faile
                                                                     value, "Address: low-level call with value faile
      d");
                                                                    d");
 114
 115
                                                               115
          function functionCallWithValue(
                                                                        function functionCallWithValue(
                                                               116
 116
              address target,
                                                               117
                                                                            address target,
 117
 118
              bytes memory data,
                                                               118
                                                                            bytes memory data,
 119
              uint256 value,
                                                               119
                                                                            uint256 value,
              string memory errorMessage
                                                                            string memory errorMessage
 120
 121
          ) internal returns (bytes memory) {
                                                                        ) internal returns (bytes memory) {
              require(address(this).balance >= value, "Ad
                                                                            require(address(this).balance >= value, "Ad
      dress: insufficient balance for call");
                                                                    dress: insufficient balance for call");
              require(isContract(target), "Address: call
                                                                            require(isContract(target), "Address: call
       to non-contract");
                                                                     to non-contract");
 124
                                                               124
              // solhint-disable-next-line avoid-low-leve
                                                               125
                                                                            // solhint-disable-next-line avoid-low-leve
 125
      1-calls
                                                                    1-calls
              (bool success, bytes memory returndata) = t
                                                                            (bool success, bytes memory returndata) = t
      arget.call{ value: value }(data);
                                                                    arget.call{ value: value }(data);
              return verifvCallResult(success, returndat
                                                                            return verifvCallResult(success, returndat
 127
      a, errorMessage);
                                                                    a, errorMessage);
128
          }
                                                               128
                                                                        }
 129
                                                               129
          function _functionCallWithValue(
                                                                        function _functionCallWithValue(
 130
                                                               130
 131
              address target,
                                                               131
                                                                            address target,
 132
              bytes memory data,
                                                               132
                                                                            bytes memory data,
 133
              uint256 weiValue,
                                                               133
                                                                            uint256 weiValue,
              string memory errorMessage
                                                               134
                                                                            string memory errorMessage
 134
          ) private returns (bytes memory) {
                                                                        ) private returns (bytes memory) {
 135
              require(isContract(target), "Address: call
                                                                            require(isContract(target), "Address: call
 136
       to non-contract");
                                                                     to non-contract");
 137
                                                               137
              // solhint-disable-next-line avoid-low-leve
                                                                            // solhint-disable-next-line avoid-low-leve
 138
                                                                    l-calls
      1-calls
              (bool success, bytes memory returndata) = t
                                                                            (bool success, bytes memory returndata) = t
 139
                                                               139
      arget.call{ value: weiValue }(data);
                                                                    arget.call{ value: weiValue }(data);
 140
              if (success) {
                                                               140
                                                                            if (success) {
 141
                  return returndata;
                                                                141
                                                                                return returndata;
              } else {
                                                                            } else {
                  // Look for revert reason and bubble it
                                                                                // Look for revert reason and bubble it
      up if present
                                                                    up if present
 144
                  if (returndata.length > 0) {
                                                               144
                                                                                if (returndata.length > 0) {
                      // The easiest way to bubble the re
                                                                                    // The easiest way to bubble the re
 145
                                                               145
      vert reason is using memory via assembly
                                                                    vert reason is using memory via assembly
                                                               146
 146
                      // solhint-disable-next-line no-inl
                                                                                     // solhint-disable-next-line no-inl
 147
                                                                147
      ine-assembly
                                                                    ine-assembly
 148
                      assembly {
                                                               148
                                                                                    assembly {
 149
                          let returndata size := mload(re
                                                               149
                                                                                         let returndata size := mload(re
      turndata)
                                                                    turndata)
 150
                          revert(add(32, returndata), ret
                                                               150
                                                                                         revert(add(32, returndata), ret
      urndata_size)
                                                                    urndata_size)
 151
                                                               151
 152
                  } else {
                                                               152
                                                                                } else {
                      revert(errorMessage);
                                                               153
                                                                                    revert(errorMessage);
 153
 154
                                                               154
 155
              }
                                                               155
                                                                            }
```

```
156
157
                                                              157
        function functionStaticCall(address target, byt
158
                                                                       function functionStaticCall(address target, byt
    es memory data) internal view returns (bytes memor
                                                                   es memory data) internal view returns (bytes memor
            return functionStaticCall(target, data, "Ad
                                                              159
                                                                           return functionStaticCall(target, data, "Ad
    dress: low-level static call failed");
                                                                   dress: low-level static call failed");
160
                                                              160
161
                                                              161
        function functionStaticCall(
                                                                       function functionStaticCall(
162
                                                              162
            address target,
                                                                           address target,
164
            bytes memory data,
                                                              164
                                                                           bytes memory data,
165
            string memory errorMessage
                                                              165
                                                                           string memory errorMessage
        ) internal view returns (bytes memory) {
                                                                       ) internal view returns (bytes memory) {
            require(isContract(target), "Address: stati
                                                                           require(isContract(target), "Address: stati
    c call to non-contract");
                                                                   c call to non-contract");
168
            // solhint-disable-next-line avoid-low-leve
                                                                           // solhint-disable-next-line avoid-low-leve
169
                                                              169
    1-calls
                                                                   1-calls
170
            (bool success, bytes memory returndata) = t
                                                              170
                                                                           (bool success, bytes memory returndata) = t
    arget.staticcall(data);
                                                                   arget.staticcall(data);
            return _verifyCallResult(success, returndat
                                                                           return _verifyCallResult(success, returndat
    a, errorMessage);
                                                                   a, errorMessage);
172
                                                              172
173
                                                              173
        function functionDelegateCall(address target, b
                                                                       function functionDelegateCall(address target, b
174
                                                              174
    ytes memory data) internal returns (bytes memory) {
                                                                   ytes memory data) internal returns (bytes memory) {
175
            return functionDelegateCall(target, data,
                                                                           return functionDelegateCall(target, data,
      "Address: low-level delegate call failed");
                                                                    "Address: low-level delegate call failed");
176
                                                              176
177
                                                              177
178
        function functionDelegateCall(
                                                              178
                                                                       function functionDelegateCall(
179
            address target,
                                                              179
                                                                           address target,
180
            bytes memory data,
                                                              180
                                                                           bytes memory data,
181
            string memory errorMessage
                                                                           string memory errorMessage
182
        ) internal returns (bytes memory) {
                                                                       ) internal returns (bytes memory) {
            require(isContract(target), "Address: deleg
                                                                           require(isContract(target), "Address: deleg
    ate call to non-contract");
                                                                   ate call to non-contract");
184
            (bool success, bytes memory returndata) = t
                                                              184
                                                                           (bool success, bytes memory returndata) = t
    arget.delegatecall(data);
                                                                   arget.delegatecall(data);
185
            return _verifyCallResult(success, returndat
                                                              185
                                                                           return _verifyCallResult(success, returndat
    a, errorMessage);
                                                                   a, errorMessage);
186
                                                              186
187
                                                              187
188
        function _verifyCallResult(
                                                              188
                                                                       function _verifyCallResult(
189
            bool success,
                                                              189
                                                                           bool success,
190
            bytes memory returndata,
                                                              190
                                                                           bytes memory returndata,
191
            string memory errorMessage
                                                              191
                                                                           string memory errorMessage
192
        ) private pure returns(bytes memory) {
                                                              192
                                                                       ) private pure returns(bytes memory) {
193
            if (success) {
                                                              193
                                                                           if (success) {
                 return returndata;
                                                                               return returndata;
195
            } else {
                                                                           } else {
                 if (returndata.length > 0) {
196
                                                              196
                                                                               if (returndata.length > 0) {
197
                                                              197
                     assembly {
                                                                                   assembly {
198
                         let returndata_size := mload(re
                                                                                       let returndata_size := mload(re
    turndata)
                                                                   turndata)
                         revert(add(32, returndata), ret
                                                                                       revert(add(32, returndata), ret
199
    urndata_size)
                                                                   urndata size)
200
                                                              200
                                                                               } else {
201
                 } else {
                                                              201
202
                     revert(errorMessage);
                                                              202
                                                                                   revert(errorMessage);
                                                              203
203
                 }
                                                                               }
204
            }
                                                              204
                                                                           }
205
                                                              205
206
                                                              206
        function addressToString(address _address) inte
                                                                       function addressToString(address _address) inte
    rnal pure returns(string memory) {
                                                                   rnal pure returns(string memory) {
208
            bytes32 _bytes = bytes32(uint256(_addres
                                                              208
                                                                           bytes32 _bytes = bytes32(uint256(_addres
    s));
                                                                   s));
```

```
bytes memory HEX = "0123456789abcdef";
                                                                          bytes memory HEX = "0123456789abcdef";
            bytes memory _addr = new bytes(42);
210
                                                              210
                                                                          bytes memory addr = new bytes(42);
                                                              211
212
             _addr[0] = '0';
                                                              212
                                                                           _addr[0] = '0';
213
            _addr[1] = 'x';
                                                              213
                                                                           _addr[1] = 'x';
                                                              214
214
            for(uint256 i = 0; i < 20; i++) {
                                                                           for(uint256 i = 0; i < 20; i++) {
215
                \_addr[2+i*2] = HEX[uint8(\_bytes[i + 12]
                                                                               \_addr[2+i*2] = HEX[uint8(\_bytes[i + 12]
    >> 4)];
                                                                  >> 4)];
217
                 \_addr[3+i*2] = HEX[uint8(\_bytes[i + 12]
                                                              217
                                                                               addr[3+i*2] = HEX[uint8(bvtes[i + 12]
    & 0x0f)];
                                                                  & 0x0f)];
218
                                                              218
219
                                                              219
            return string(_addr);
                                                              220
                                                                           return string(_addr);
220
                                                              221
222
                                                              222
223 }
                                                              223 }
224
                                                              224
225 contract OwnableData {
                                                              225 contract OwnableData {
226
        address public owner;
                                                                      address public owner;
227
        address public pendingOwner;
                                                                      address public pendingOwner;
                                                              227
228 }
                                                              228 }
                                                              229
230 contract Ownable is OwnableData {
                                                              230 contract Ownable is OwnableData {
        event OwnershipTransferred(address indexed prev
                                                                      event OwnershipTransferred(address indexed prev
    iousOwner, address indexed newOwner);
                                                                  iousOwner, address indexed newOwner);
232
                                                              232
        /// @notice `owner` defaults to msg.sender on c
                                                                      /// @notice `owner` defaults to msg.sender on c
    onstruction.
                                                                  onstruction.
234
        constructor() {
                                                              234
                                                                    constructor() {
235
            owner = msg.sender;
                                                                          owner = msg.sender;
236
            emit OwnershipTransferred(address(0), msg.s
                                                                          emit OwnershipTransferred(address(0), msg.s
    ender);
                                                                  ender);
237
                                                              237
238
        /// @notice Transfers ownership to `newOwner`.
                                                              239
                                                                      /// @notice Transfers ownership to `newOwner`.
     Either directly or claimable by the new pending ow
                                                                   Either directly or claimable by the new pending ow
        /// Can only be invoked by the current `owner`.
                                                              240
                                                                    /// Can only be invoked by the current `owner`.
        /// @param newOwner Address of the new owner.
241
                                                              241
                                                                      /// @param newOwner Address of the new owner.
        /// @param direct True if `newOwner` should be
                                                                      /// @param direct True if `newOwner` should be
242
     set immediately. False if `newOwner` needs to use
                                                                   set immediately. False if `newOwner` needs to use
     `claimOwnership`.
                                                                    `claimOwnership`
        /// @param renounce Allows the `newOwner` to be
                                                                      /// @param renounce Allows the `newOwner` to be
     `address(0)` if `direct` and `renounce` is True. Ha
                                                                   `address(0)` if `direct` and `renounce` is True. Ha
    s no effect otherwise.
                                                                  s no effect otherwise.
244
        function transferOwnership(
                                                              244
                                                                      function transferOwnership(
            address newOwner.
                                                                          address newOwner.
245
                                                              245
246
            bool direct.
                                                              246
                                                                          bool direct,
247
            bool renounce
                                                              247
                                                                          bool renounce
        ) public onlyOwner {
                                                                      ) public onlyOwner {
248
                                                              248
            if (direct) {
                                                                           if (direct) {
250
                 // Checks
                                                                               // Checks
251
                require(newOwner != address(0) || renou
                                                              251
                                                                              require(newOwner != address(0) || renou
    nce, "Ownable: zero address");
                                                                  nce, "Ownable: zero address");
                                                              252
253
                 // Effects
                                                              253
                                                                               // Effects
254
                 emit OwnershipTransferred(owner, newOwn
                                                              254
                                                                               emit OwnershipTransferred(owner, newOwn
                 owner = newOwner;
                                                              255
                                                                               owner = newOwner;
256
                pendingOwner = address(0);
                                                                              pendingOwner = address(0);
257
            } else {
                                                              257
                                                                          } else {
                 // Effects
                                                                               // Effects
258
                                                              258
                 pendingOwner = newOwner;
                                                                               pendingOwner = newOwner;
260
                                                              260
261
                                                              261
262
                                                              262
        /// @notice Needs to be called by `pendingOwner
                                                                      /// @notice Needs to be called by `pendingOwner
                                                                  ` to claim ownership.
     ` to claim ownership.
264
        function claimOwnership() public {
                                                              264
                                                                      function claimOwnership() public {
```

209

```
265
            address _pendingOwner = pendingOwner;
                                                                        address _pendingOwner = pendingOwner;
266
                                                            266
            // Checks
                                                                        // Checks
267
                                                            267
            require(msg.sender == _pendingOwner, "Ownab
                                                            268
                                                                        require(msg.sender == _pendingOwner, "Ownab
    le: caller != pending owner");
                                                                le: caller != pending owner");
269
                                                            269
            // Effects
                                                            270
                                                                        // Effects
270
271
            emit OwnershipTransferred(owner, _pendingOw
                                                                        emit OwnershipTransferred(owner, _pendingOw
    ner);
                                                                ner);
272
            owner = _pendingOwner;
                                                            272
                                                                        owner = pendingOwner;
            pendingOwner = address(0);
                                                                        pendingOwner = address(0);
                                                            273
274
                                                            274
275
                                                            275
        /// @notice Only allows the `owner` to execute
                                                                    /// @notice Only allows the `owner` to execute
276
     the function.
                                                                 the function.
277
        modifier onlyOwner() {
                                                            277
                                                                    modifier onlyOwner() {
278
            require(msg.sender == owner, "Ownable: call
                                                                        require(msg.sender == owner, "Ownable: call
                                                            278
    er is not the owner");
                                                                er is not the owner");
                                                            279
280
                                                            280
281
                                                            281
                                                            282
283 interface ITreasury {
                                                            283 interface ITreasury {
        function mintRewards( address _recipient, uint
                                                                    function mintRewards( address _recipient, uint
284
                                                            284
      _amount ) external;
                                                                 _amount ) external;
285 }
                                                            285 }
286
                                                            286
    contract Distributor is Ownable {
                                                            287
                                                                contract Distributor is Ownable {
287
        using LowGasSafeMath for uint;
                                                                    using LowGasSafeMath for uint;
288
                                                            288
289
        using LowGasSafeMath for uint32;
                                                            289
                                                                    using LowGasSafeMath for uint32;
290
                                                            290
291
                                                            291
292
                                                            292
        293
                                                            293
294
                                                            294
295
        IERC20 public immutable TIME;
                                                            295
                                                                    IERC20 public immutable TIME;
        ITreasury public immutable treasury;
                                                                    ITreasury public immutable treasury;
296
                                                            296
297
                                                            297
298
        uint32 public immutable epochLength;
                                                                    uint32 public immutable epochLength;
                                                            298
        uint32 public nextEpochTime;
                                                                    uint32 public nextEpochTime;
299
300
                                                            300
301
        mapping( uint => Adjust ) public adjustments;
                                                            301
                                                                    mapping( uint => Adjust ) public adjustments;
                                                            302
302
        event LogDistribute(address indexed recipient,
                                                                    event LogDistribute(address indexed recipient,
303
                                                            303
     uint amount);
                                                                 uint amount):
304
        event LogAdjust(uint initialRate, uint currentR
                                                            304
                                                                    event LogAdjust(uint initialRate, uint currentR
    ate, uint targetRate);
                                                                ate, uint targetRate);
305
        event LogAddRecipient(address indexed recipien
                                                            305
                                                                    event LogAddRecipient(address indexed recipien
    t, uint rate);
                                                                t, uint rate);
306
        event LogRemoveRecipient(address indexed recipi
                                                            306
                                                                    event LogRemoveRecipient(address indexed recipi
307
                                                            307
        308
                                                            308
                                                                    309
                                                            309
310
        struct Info {
                                                            310
                                                                    struct Info {
            uint rate; // in ten-thousandths ( 5000 =
                                                                        uint rate; // in ten-thousandths ( 5000 =
311
                                                            311
     0.5%)
                                                                 0.5%)
312
            address recipient;
                                                            312
                                                                        address recipient;
313
                                                            313
314
        Info[] public info;
                                                            314
                                                                    Info[] public info;
315
                                                            315
316
        struct Adjust {
                                                            316
                                                                    struct Adjust {
            bool add:
                                                                        bool add:
317
                                                            317
318
            uint rate;
                                                            318
                                                                        uint rate;
            uint target;
                                                                        uint target;
        }
321
                                                            321
                                                            322
322
323
                                                            323
```

```
324
        /* ===== CONSTRUCTOR ===== */
                                                              324
                                                                       /* ===== CONSTRUCTOR ===== */
325
                                                              325
326
        constructor( address _treasury, address _time,
                                                              326
                                                                       constructor( address _treasury, address _time,
     uint32 _epochLength, uint32 _nextEpochTime ) {
                                                                    uint32 _epochLength, uint32 _nextEpochTime ) {
327
            require( _treasury != address(0) );
                                                              327
                                                                           require( _treasury != address(0) );
            treasury = ITreasury(_treasury);
                                                                           treasury = ITreasury(_treasury);
            require( _time != address(0) );
                                                                           require( _time != address(0) );
330
            TIME = IERC20(_time);
                                                                           TIME = IERC20(_time);
            {\sf epochLength} \; = \; \_{\sf epochLength};
                                                              331
                                                                           epochLength = _epochLength;
331
332
            nextEpochTime = _nextEpochTime;
                                                              332
                                                                           nextEpochTime = _nextEpochTime;
333
        }
                                                              333
                                                                       }
334
                                                              334
335
                                                              335
        /* ===== PUBLIC FUNCTIONS ====== */
                                                                       /* ===== PUBLIC FUNCTIONS ===== */
337
338
339
                                                              339
                                                                           @notice send epoch reward to staking contra
340
            @notice send epoch reward to staking contra
341
                                                              341
        function distribute() external returns ( bool )
                                                                       function distribute() external returns ( bool )
343
            if ( nextEpochTime <= uint32(block.timestam</pre>
                                                              343
                                                                           if ( nextEpochTime <= uint32(block.timestam</pre>
    p)){
                                                                   p)){
344
                nextEpochTime = nextEpochTime.add32( ep
                                                              344
                                                                               nextEpochTime = nextEpochTime.add32( ep
    ochLength ); // set next epoch time
                                                                   ochLength ); // set next epoch time
345
                                                              345
                 // distribute rewards to each recipient
                                                                               // distribute rewards to each recipient
346
                                                              346
                 for ( uint i = 0; i < info.length; i++
                                                                               for ( uint i = 0; i < info.length; i++
348
                     if ( info[ i ].rate > 0 ) {
                                                                                   if ( info[ i ].rate > 0 ) {
                                                              348
                         treasury.mintRewards( // mint a
                                                                                       treasury.mintRewards( // mint a
349
    nd send from treasury
                                                                   nd send from treasury
350
                             info[ i ].recipient,
                                                              350
                                                                                           info[ i ].recipient,
351
                             nextRewardAt( info[ i ].rat
                                                                                           nextRewardAt( info[ i ].rat
352
                                                              352
                         adjust( i ); // check for adjus
                                                                                       adjust( i ); // check for adjus
353
                                                              353
    tment
                                                                   tment
354
                                                              354
355
                     emit LogDistribute(info[ i ].recipi
                                                                                   emit LogDistribute(info[ i ].recipi
                                                                   ent, nextRewardAt( info[ i ].rate ));
    ent, nextRewardAt( info[ i ].rate ));
                                                              357
357
                return true;
                                                                               return true;
358
            } else {
                                                              358
                                                                           } else {
                 return false;
                                                                               return false;
359
                                                              359
360
                                                              360
361
                                                              361
        /* ===== INTERNAL FUNCTIONS ====== */
                                                                       /* ===== INTERNAL FUNCTIONS ====== */
365
                                                              365
367
                                                              367
368
            @notice increment reward rate for collector
                                                              368
                                                                           @notice increment reward rate for collector
                                                              369
370
        function adjust( uint _index ) internal {
                                                              370
                                                                       function adjust( uint _index ) internal {
            Adjust memory adjustment = adjustments[ _in
                                                                           Adjust memory adjustment = adjustments[ _in
    dex 1;
                                                                   dex 1;
372
            if ( adjustment.rate != 0 ) {
                                                              372
                                                                           if ( adjustment.rate != 0 ) {
                                                                               uint initial = info[ _index ].rate;
373
                 uint initial = info[ _index ].rate;
                                                              373
374
                 uint rate = initial;
                                                              374
                                                                               uint rate = initial;
                if ( adjustment.add ) { // if rate shou
                                                                               if ( adjustment.add ) { // if rate shou
                                                                   ld increase
                     rate = rate.add( adjustment.rate );
                                                                                   rate = rate.add( adjustment.rate );
                                                                   // raise rate
    // raise rate
                     if ( rate >= adjustment.target ) {
                                                                                   if ( rate >= adjustment.target ) {
     // if target met
                                                                    // if target met
```

```
378
                        rate = adjustment.target;
                                                             378
                                                                                      rate = adjustment.target;
                        delete adjustments[ _index ];
                                                                                     delete adjustments[ _index ];
379
                                                             379
380
                    }
                                                             380
                                                                                 }
381
                } else { // if rate should decrease
                                                             381
                                                                             } else { // if rate should decrease
382
                    rate = rate.sub( adjustment.rate );
                                                                                  rate = rate.sub( adjustment.rate );
                    if ( rate <= adjustment.target ) {</pre>
                                                                                  if ( rate <= adjustment.target ) {</pre>
     // if target met
                                                                  // if target met
                        rate = adjustment.target;
                                                                                      rate = adjustment.target;
384
                                                             384
                        delete adjustments[ _index ];
                                                                                      delete adjustments[ _index ];
                                                             385
386
                                                             386
387
                                                             387
                info[ _index ].rate = rate;
                                                                              info[ _index ].rate = rate;
388
                                                             388
                emit LogAdjust(initial, rate, adjustmen
                                                                              emit LogAdjust(initial, rate, adjustmen
    t.target);
                                                                  t.target);
390
                                                             390
            }
                                                                         }
                                                             391
        }
                                                                     }
392
                                                             392
393
                                                             393
394
                                                             394
395
        396
                                                             396
397
                                                             397
398
            @notice view function for next reward at gi
                                                             398
                                                                         @notice view function for next reward at gi
    ven rate
                                                                  ven rate
399
            @param _rate uint
                                                             399
                                                                         @param _rate uint
400
                                                             400
            @return uint
                                                                          @return uint
401
                                                             401
        function nextRewardAt( uint _rate ) public view
                                                                     function nextRewardAt( uint _rate ) public view
    returns ( uint ) {
                                                                  returns ( uint ) {
            return TIME.totalSupply().mul( _rate ).div(
                                                                         return TIME.totalSupply().mul( _rate ).div(
403
                                                             403
                                                                 1000000);
    1000000);
404
        }
                                                             404
405
                                                             405
                                                             406
406
            @notice view function for next reward for s
                                                                          @notice view function for next reward for s
    pecified address
                                                                 pecified address
408
            @param recipient address
                                                             408
                                                                         @param recipient address
409
            @return uint
                                                             409
                                                                         @return uint
410
                                                             410
411
        function nextRewardFor( address _recipient ) ex
                                                             411
                                                                     function nextRewardFor( address _recipient ) ex
     ternal view returns ( uint ) {
                                                                  ternal view returns ( uint ) {
412
            uint reward;
                                                             412
                                                                         uint reward;
413
            for ( uint i = 0; i < info.length; i++ ) {
                                                             413
                                                                          for ( uint i = 0; i < info.length; i++ ) {
                if ( info[ i ].recipient == recipient
                                                             414
                                                                             if ( info[ i ].recipient == recipient
414
     ) {
                                                                  ) {
415
                    reward = nextRewardAt( info[ i ].ra
                                                             415
                                                                                  reward = nextRewardAt( info[ i ].ra
    te );
                                                                  te );
416
                                                             416
            }
417
                                                             417
418
            return reward;
                                                             418
                                                                          return reward;
419
        }
                                                             419
                                                                     }
420
                                                             420
421
                                                             421
422
        /* ===== POLICY FUNCTIONS ====== */
                                                                      /* ===== POLICY FUNCTIONS ====== */
423
                                                             423
424
                                                             424
            @notice adds recipient for distributions
                                                                          Onotice adds recipient for distributions
426
                                                             426
427
            @param recipient address
                                                             427
                                                                          @param recipient address
            @param _rewardRate uint
                                                                          @param _rewardRate uint
428
                                                             428
429
                                                             429
        function addRecipient( address _recipient, uint
                                                                      function addRecipient( address _recipient, uint
430
                                                             430
    _rewardRate ) external onlyOwner {
                                                                  _rewardRate ) external onlyOwner {
            require( _recipient != address(0), "IA" );
                                                             431
                                                                          require( _recipient != address(0), "IA" );
            require(_rewardRate <= 5000, "Too high rewa
                                                             432
                                                                          require(_rewardRate <= 50000, "Too high rew</pre>
     rd rate");
                                                                  ard rate");
```

```
433
            require(info.length <= 4, "limit recipients</pre>
                                                              433
                                                                           require(info.length <= 4, "limit recipients
    max to 5");
                                                                   max to 5");
434
            info.push( Info({
                                                              434
                                                                           info.push( Info({
435
                recipient: _recipient,
                                                              435
                                                                               recipient: _recipient,
436
                rate: _rewardRate
                                                              436
                                                                               rate: _rewardRate
437
                                                              437
             emit LogAddRecipient(_recipient, _rewardRat
                                                                            emit LogAddRecipient(_recipient, _rewardRat
    e);
                                                                   e);
439
        }
                                                              439
                                                                       }
440
                                                              440
441
                                                              441
442
             \\ @ notice \ removes \ recipient \ for \ distributions
                                                              442
                                                                            @notice removes recipient for distributions
443
             @param _index uint
                                                              443
                                                                            @param _index uint
             @param _recipient address
                                                                            @param _recipient address
                                                              444
445
                                                              445
        function removeRecipient( uint _index, address
                                                                       function removeRecipient( uint _index, address
      _recipient ) external onlyOwner {
                                                                     recipient ) external onlyOwner {
            require( _recipient == info[ _index ].recip
                                                                           require( _recipient == info[ _index ].recip
                                                              447
447
    ient, "NA" );
                                                                   ient, "NA" );
448
            info[_index] = info[info.length-1];
                                                              448
                                                                           info[_index] = info[info.length-1];
             adjustments[_index] = adjustments[ info.len
                                                                           adjustments[_index] = adjustments[ info.len
    gth-1 ];
                                                                   gth-1 ];
450
                                                              450
             info.pop();
                                                                            info.pop();
451
             delete adjustments[ info.length-1 ];
                                                              451
                                                                           delete adjustments[ info.length-1 ];
452
             emit LogRemoveRecipient(_recipient);
                                                              452
                                                                            emit LogRemoveRecipient(_recipient);
453
        }
                                                              453
                                                                       }
454
                                                               454
455
                                                               455
             @notice set adjustment info for a collecto
                                                                            @notice set adjustment info for a collecto
      's reward rate
                                                                     's reward rate
                                                              457
457
            @param _index uint
                                                                           @param _index uint
458
             @param _add bool
                                                              458
                                                                            @param _add bool
459
             @param _rate uint
                                                              459
                                                                           @param _rate uint
460
             @param _target uint
                                                              460
                                                                           @param _target uint
461
                                                              461
        function setAdjustment( uint _index, bool _add,
                                                                       function setAdjustment( uint _index, bool _add,
    uint _rate, uint _target ) external onlyOwner {
                                                                   uint _rate, uint _target ) external onlyOwner {
             require(_target <= 5000, "Too high reward r</pre>
                                                                            require(_target <= 50000, "Too high reward
463
                                                              463
                                                                    rate");
    ate");
             adjustments[ _index ] = Adjust({
                                                                            adjustments[ _index ] = Adjust({
464
                                                              464
465
                 add: _add,
                                                              465
                                                                                add: _add,
466
                 rate: _rate,
                                                               466
                                                                                rate: _rate,
                 target: _target
                                                               467
                                                                                target: _target
468
            });
                                                              468
                                                                           });
469
        }
                                                              469
                                                                       }
470 }
                                                              470 }
```