Halborn-1

Improper key management policy: It is recommended to include multi-signature in the key management policy.

Pera-1

We plan to deploy a multisig wallet via Gnosis in order to be able to use the contract owner specific (privileged) functions. Policy will be as follows:

1- PERA smart contract deployment on BSC.
2- Creating a multisig wallet on Gnosis where it requires at least 3 out of 4 confirmation to access the wallet.
3- Transferring the ownership of the PERA smart contract to the multisig wallet.

Details will be discussed and implemented with the Halborn Team.

Halborn-2

Improper role-based access control: Recommendable to use role-based access control

Pera-2

Issue will be resolved when the contract ownership is transferred to a multisig wallet.

Halborn-3

No test coverage: Perform as much as possible test cases to cover all possible scenarios

Pera-3

We have made extensive testing for checking the algo and math behind the each feature and function. On the next page you can find one of these tests where we tested the holder reward distribution. We ran similar tests for LP token staker reward distribution, pool share calculations, trading competition sorting algorithm, trading competition reward distributions etc. However these tests have been made manually, without the use of a test script. Later this week (12.04.2021 – 19.04.2021), we planned to automate the manual tests with test suites and add the codes to to Pera GitHub repository. We will update the Halborn Team as soon as the tests are ready.

Halborn-4

Floating Pragma: Consider lock the pragma version known bugs for the compiler version.

Pera-4

Fixed

Pera-3

Holder reward distribution test results

		Rate	1										
			2.00%	10,000,000									
Transaction			0.75%	Contract Deployer	PancakeSwap	Wallet-2	Wallet-3	Wallet-4	Wallet-5	Wallet-6	Wallet-7	Wallet-8	
	Tx Amount		0	6,000,000	4,000,000	0.0000							Included Supply
Buy-2	Fee		0			0.9924							0.000000
			0			0.000000000009875							
	Tx Amount		1,000	6,000,000	3,999,000	973							Included Supply
Buy-2	Fee		20			0.9849							988
			8			987.5000							
	Tx Amount		2,444.4		3,996,555.6	973	2,359.2627						Included Supply
Buy-3	Fee		49			0.9796	0.9796						3,401.35
			18			992.8514	2,408.4936						
	Tx Amount		5,120.0	6,000,000	3,991,435.6	973	2,359	4,915.0375					Included Supply
Buy-4	Fee		102			0.9751	0.9751	0.9751					8,457.35
			38			997.3799	2,419.4791	5,040.48598					
	Tx Amount		1,111.0	6,000,000	3,990,324.6	973	2,359	4,915.037468	1,061.6822				Included Supply
Buy-5	Fee		22			0.9743	0.9743	0.9743	0.9743				9,554.458
			8			998.250512247	2,421.590968197	5,044.8857	1,089.73036041849				
	Tx Amount		555	6,000,000	3,990,324.6	432	2,359.2627	4,915.0375	1,061.6822	529.9008			Included Supply
Transfer 2-6	Fee		11			0.9738	0.9738	0.9738	0.9738	0.9738			9,548
			4			443.443843604	2,422.64718687	5,047.0861	1,090.205665113	544.137231367			
	Tx Amount		621.83898440	6,000,000	3,990,324.6000	431.8418737	2,359.2627	5,520.6071	1,061.6822	529.9008			Included Supply
Claim TC – 4	Fee		0.0000			0.9738	0.9738	0.9738	0.9738	0.9738			10,169
			0.0000			443.443843604	2,422.64718687	5,668.9251	1,090.205665113	544.137231367			
	Tx Amount		2,500	6,000,000	3,992,774.6	432	2,359.2627	3,086.0154	1,061.6822	529.9008			Included Supply
Sell-4	Fee		50			0.9715	0.9715	0.9715	0.9715	0.9715			7,688
			19			444.527972289	2,428.57005928	3,176.67242524	1,092.870992980	545.467534605186			
	Tx Amount		750	6,000,000	3,993,509.6	432	2,359.2627	3,086.0154	333.0860	529.9008			Included Supply
Add Liq-5	Fee		15			0.9707	0.9707	0.9707	0.9707	0.9707			6,944
			6			444.888368739	2,430.53899731	3,179.24787951	343.148972087	545.909766759			
	Tx Amount		1,500	6,000,000	3,994,979.6	432	903.2506	3,086.0154	333.0860	529.9008			Included Supply
Sell-3	Fee		30			0.9687	0.9687	0.9687	0.9687	0.9687			5,455
			11			445.807773354	932.46204567	3,185.81810112	343.858122450	547.037941812			
	Tx Amount		932	6,000,000	3,995,893.41280475	431.8419	0.0000	3,086.0154	333.0860	529.9008			Included Supply
Sell-3	Fee		19			0.9672	0.9672	0.9672	0.9672	0.9672			4,530
			7			446.497154468	0.00000000	3,190.74453570	344.389852289	547.883861620			
	Tx Amount		1,000	6,000,000	3,994,893.41280476	431.8419	947.8337	3,086.0154	333.0860	529.9008			Included Supply
Buy-3	Fee		20			0.9659	0.9659	0.9659	0.9659	0.9659			5,517
			8			447.104962671	981.33405562	3,195.08803640	344.858663737	548.629685646			
	Tx Amount		367.627	6,000,000	3,994,526	432	947.8337	3,086.0154	333.0860	529.9008			Included Supply
Rem Liq-5	Fee		7			0.9654	0.9654	0.9654	0.9654	0.9654			5,520
			3			447.328409539	981.82449084	3,196.68482569	345.031011608	548.903870894			
	Tx Amount		360.274721194	6,000,000	3,994,525.78553823	432	947.8337	3,086.0154	673.9320	529.9008			Included Supply
Rem Liq-5	Fee		7			0.9649	0.9649	0.9649	0.9649	0.9649			5,876
			3			447.534222742	982.276222588	3,198.155601804	698.421430242	549.156418376			
	Tx Amount		3,333	6,000,000	3,991,192.78553823	432	947.8337	3,086.0154	673.9320	529.9008	3,151.8090		Included Supply
Buy-7	Fee		67			0.9623	0.9623	0.9623	0.9623	0.9623	0.9623		9,166.88
			25			448.757957024	984.96215146	3,206.90061493	700.3311886577110	550.658027642	3,275.271456041		
	Block #												Included Supply
LP Stake – 2	7851575												
	Tx Amount		3,333.333333	6,000,000	3,987,859.45220523	432	947.8337	3,086.0154	673.9320	529.9008	3,151.8090	3,143.5286	Included Supply
Buy-8	Fee		66.66667			0.9604	0.9604	0.9604	0.9604	0.9604	0.9604	0.9604	12,459
			25.00000			449.660269772 9	986.94260415	3,213.34869515	701.739336970	551.765229756	3,281.857008772	3,273.234917514	
	Block #		45.5000	6,000,000	3,987,859.45220523	475.5389	947.8337	3,086.0154	673.9320	529.9008	3,151.8090	3,143.5286	Included Supply
LP Unstake – 2	7851616		0			0.9604	0.9604	0.9604	0.9604	0.9604	0.9604	0.9604	12,504
			0			495.1602698	986.9426042	3,213.3486952	701.7393370	551.7652298	3,281.8570088	3,273.2349175	
	Tx Amount		2,151.5	6,000,000	3,987,859.45220523	476	947.8337	1,019.7715	673.9320	529.9008	3,151.8090	5,168.4476	Included Supply
Transfer 4-8	Fee		43.03			0.9591	0.9591	0.9591	0.9591	0.9591	0.9591	0.9591	12,477
			16.14			495.8014718	988.220633967	1,063.2237237357800	702.6480461	552.4797317	3,286.1068112	5,388.6738935	
	Tx Amount		959.27974	6,000,000	3,987,859.45220523	476	1,867.9092	1,019.7715	673.9320	529.9008	3,151.8090	5,168.4476	Included Supply
Claim TC – 3	Fee		0			0.9591	0.9591	0.9591	0.9591	0.9591	0.9591	0.9591	13,436
			0			495.8014718	1,947.500373967	1,063.2237237357800	702.6480461	552.4797317	3,286.1068112	5,388.6738935	
	Tx Amount		1,255.0	6,000,000	3,986,604.45220523	476	3,047.5452	1,019.771505	673.9320	529.9008	3,151.8090	5,168.4476	Included Supply
Buy-3	Fee		25			0.9585	0.9585	0.9585	0.9585	0.9585	0.9585	0.9585	14,675.747
			9			496.119665231	3,179.439553009	1,063.906075117	703.09898873411	552.83429989140	3,288.21575544295	5,392.13221466163	
	Block #												Included Supply
LP Stake – 2	7851776												
	Tx Amount		1,110.0	6,000,000	3,985,494.45220523	1,518.213100	3,047.545205	1,019.771505	673.931967	529.900787	3,151.808988	5,168.447584	Included Supply
Buy-2	Fee		22			0.9580	0.9580	0.9580	0.9580	0.9580	0.9580	0.9580	15,771.872
			8			1,584.756160443	3,181.118669710	1,064.467942210	703.47030740037	553.12626119157	3,289.95231872670	5,394.97989240589	
	Block #												Included Supply
LP Stake – 3	7851845												
	Tx Amount		1,000.0	6,000,000	3,986,474.5	1,518.213100	2,089.534668	1,019.771505	673.931967	529.900787	3,151.808988	5,168.447584	Included Supply
Sell – 3	Fee		20			0.9575	0.9575	0.9575	0.9575	0.9575	0.9575	0.9575	14,779.372
			8			1,585.560775563	2,182.226071028	1,065.008395702	703.82747455329	553.40709540708	3,291.62269896955	5,397.71904086453	
	Block #		59.0750	6,000,000	3,986,474.45220523	1,574.7789	2,089.5347	1,019.7715	673.9320	529.9008	3,151.8090	5,168.4476	Included Supply
LP Unstake – 2	7851895		0			0.9575	0.9575	0.9575	0.9575	0.9575	0.9575	0.9575	14838.44655
HALF			0			1,644.635775563	2,182.226071028	1,065.008395702	703.82747455329	553.40709540708	3,291.62269896955	5,397.71904086453	
	Tx Amount		999	6,000,000	3,985,475.45220523	1,575	2,089.5347	1,019.7715	673.9320	529.9008	4,089.2445	5,168.4476	Included Supply
Buy-7	Fee		20			0.9571	0.9571	0.9571	0.9571	0.9571	0.9571	0.9571	15,824.96
			7			1,645.414815209	2,183.259759247	1,065.512874430	704.16086721992	553.66923616449	4,272.66563926259	5,400.27586055373	
	Block #		154.57833333	6,000,000	3,985,475.45220523	1,574.7789	2,237.4771	1,019.7715	673.9320	529.9008	4,089.2445	5,168.4476	Included Supply
LP Unstake – 3	7852295		0			0.9571	0.9571	0.9571	0.9571	0.9571	0.9571	0.9571	15,980
			0			1,645.414815209	2,337.838092581	1,065.512874430	704.16086721992	553.66923616449	4,272.66563926259	5,400.27586055373	
	Tx Amount		1,000	6,000,000	3,986,455.5	1,575	1,280.4061	1,019.7715	673.9320	529.9008	4,089.2445	5,168.4476	Included Supply
Add Liq-3	Fee		20			0.9566	0.9566	0.9566	0.9566	0.9566	0.9566	0.9566	14,987
			8			1,646.238646466	1,338.507925396	1,066.046358633	704.51342860871	553.94644894913	4,274.80489039365	5,402.97968697454	
	Block #		150.1642	6,000,000	3,986,455.45220523	1,718.4247	1,280.4061	1,019.7715	673.9320	529.9008	4,089.2445	5,168.4476	Included Supply
LP Unstake – 2	7852442		0			0.9566	0.9566	0.9566	0.9566	0.9566	0.9566	0.9566	15,137
			0			1,796.402813133	1,338.507925396	1,066.046358633	704.51342860871	553.94644894913	4,274.80489039365	5,402.97968697454	
	Tx Amount		980.000	6,000,000	3,985,475.5	1,718.42471	1,280.40608	1,019.77151	673.93197	529.90079	4,089.24451	5,168.44758	Included Supply
Rem Liq-3	Fee		20			0.9561	0.9561	0.9561	0.9561	0.9561	0.9561	0.9561	15,145
			7			1,797.275072159	1,339.157849572	1,066.563986723	704.85551123756	554.21542247014	4,276.88055912517	5,405.60315080057	
	Tx Amount		960.400	6,000,000	3,985,475.45220523	1,718	2,180.3059	1,019.7715	673.9320	529.9008	4,089.2445	5,168.4476	Included Supply
Rem Liq-3	Fee		19			0.9557	0.9557	0.9557	0.9557	0.9557	0.9557	0.9557	16,092.946552088
			7			1,798.079870040970	2,281.370962455	1,067.041581083	705.17113694867	554.46359338466	4,278.79569413696	5,408.02371403839	

Halborn-5

Possible Re-entrancy: External calls should be at the end of the function

Pera-5

Fixed

```
644
645     // Funciton for trading competition winners to claim their rewards
646     function getTCreward(uint _bnum) external {
647         require(_bnum > 0,"min 1 ended TC is required.");
648         require(_bnum.sub(1) < showBnum(), 'At least 1 day is Required!');
649         (uint256 _traderReward, uint256 _traderRewardEligible, uint _winnerIndex, uint256 _rewardEmission, uint256 _rewardFee) =
650         require(_traderRewardEligible > 0, 'No Eligible Reward!');
651         if(_winnerIndex != 404) {
652         isPaid[nMixAddrandSpBlock(msg.sender, _bnum.sub(1))] = true;
653         _mint(msg.sender, _rewardEmission);
654         _transfer(address(this), msg.sender, _traderRewardEligible);
655         }
656     }
```

```
720     // Function for staking LP tokens (min 1 LP token is required)
721     function depositLPtoken(uint256 _amount) external {
722
723         LPUserInfo storage user = userInfo[msg.sender];
724         updateRate(totalStakedLP);
725
726         if (user.userLPamount > 0) {
727             uint256 pendingReward = user.userLPamount.mul(LPRate).div(1e12).sub(user.userReflectedLP);
728             if(pendingReward > 0) {
729                 _transfer(address(this), msg.sender, pendingReward);
730             }
731         }
732         if (_amount > 1 * 10 ** LPTokenDecimals) {
733             totalStakedLP += _amount;
734             user.userLPamount = user.userLPamount.add(_amount);
735             BEP20(lpTokenAddress).transferFrom(msg.sender, address(this), _amount);
736         }
737         user.userReflectedLP = user.userLPamount.mul(LPRate).div(1e12);
738     }
739
```

```
770     // Function is used to withdraw LP tokens from the PERA smart contract
771     function withdraw(uint256 _amount) public {
772
773         LPUserInfo storage user = userInfo[msg.sender];
774         require(user.userLPamount >= _amount, "withdraw: not good");
775         updateRate(totalStakedLP);
776
777         uint256 pendingReward = user.userLPamount.mul(LPRate).div(1e12).sub(user.userReflectedLP);
778         if(pendingReward > 0) {
779             _transfer(address(this), msg.sender, pendingReward);
780         }
781         if(_amount > 0) {
782             user.userLPamount = user.userLPamount.sub(_amount);
783             totalStakedLP -= _amount;
784             BEP20(lpTokenAddress).transfer(msg.sender,  _amount);
785         }
786         if(totalStakedLP == 0){
787             FeeRewPoolLP = 0;
788         }
789         user.userReflectedLP = user.userLPamount.mul(LPRate).div(1e12);
790     }
```

Halborn-6

Divide before multiple: Consider doing multiplication operation before division to prevail precision in the values in non floating data type.

Pera-6

Calculations have been fixed.

Halborn-7

Pragma Version: It is suggested to switch to pragma 0.6.12

Pera-7

Fixed

Halborn-8

For loop over dynamic array: If you absolutely must loop over an array of unknown size, then you should plan for it to potentially take multiple blocks, and therefore require multiple transactions.

Pera-8

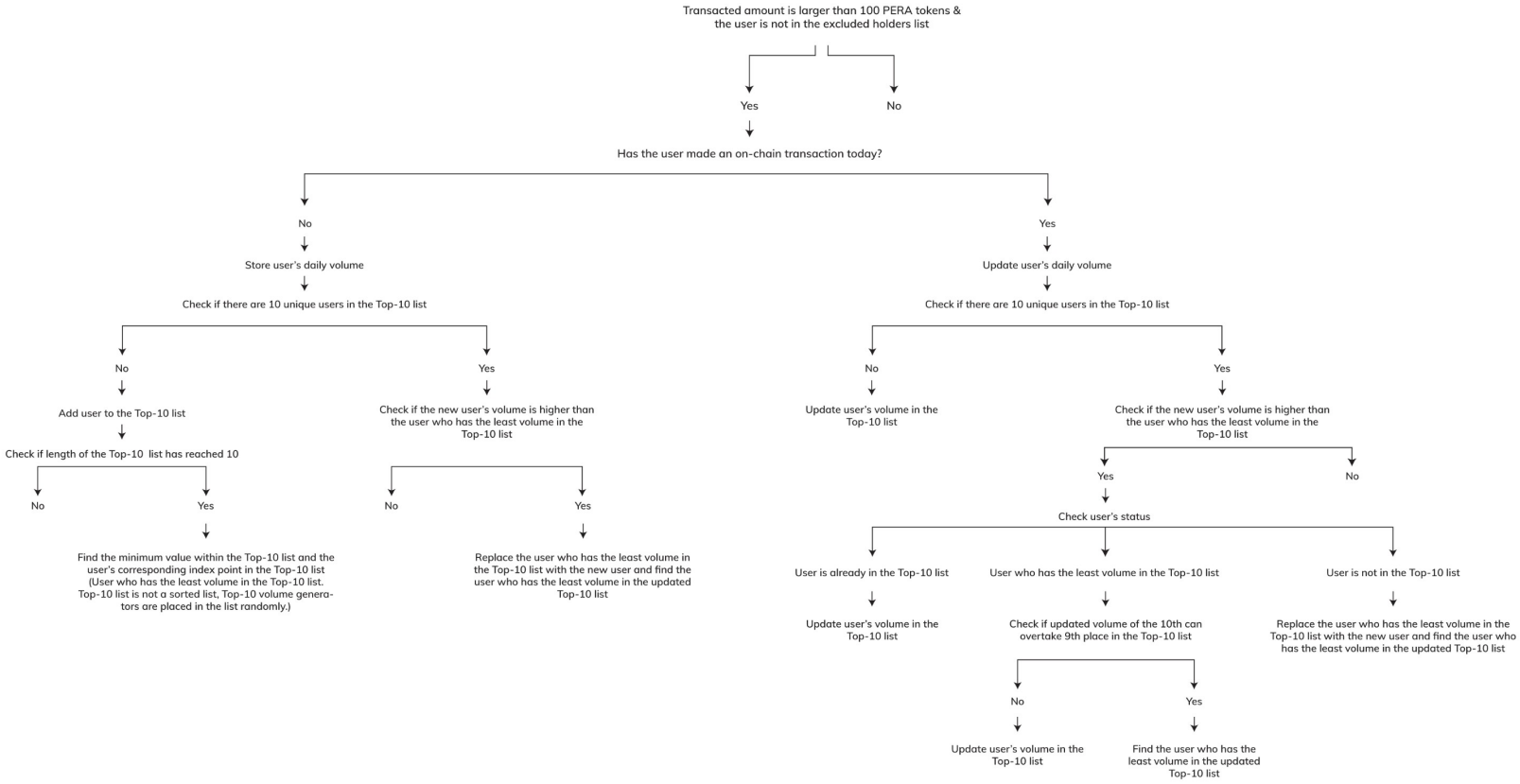
Any for loop that runs through an array whose length is unknown has been removed. In the current contract, for loops are designed to run no more than 100 loops. For loop issue was related to both trading competition sorting algorithm and LP staker reward distribution.

For the trading competition, we have designed a novel sorting algorithm that uses an unsorted list where user's placed within if their daily volume is higher than the minimum value within the list. Unsorted Top-10 traders list is only sorted when a user claims its trading competition rewards. In this case, the maximum length that a for loop runs is 100.

Diagram below explains the algorithm that we applied for the trading competition. We have also made tests to see if the gas requirement inflates when the number of unique daily traders increases. Our tests showed that the gas requirement of the smart contract is limited and the transaction gas fees are quite low.

For the LP staker reward distribution, we decided to employ a mechanism that is similar to MasterChef smart contract. It solved our issue with the for loops that runs through an array whose size is unknown.

PERA Trading Competition Sorting Algorithm Map



Halborn-9

Possible misuse of public functions: All can access to public functions while external functions only can be accessed externally.

Pera-9

Fixed

```
209     function excludeAccount(address account) external {
210         require(msg.sender == manager);
211         require(!_isExcluded(account));
212         _excluded.push(account);
213         userbalanceOf[account] = userbalanceOf[account].div(transferRate);
214     }
215
216     // Function can only be used by the contract owner
217     // Used for removing an address from the excluded holders list
218     function includeAccount(address account) external {
219     require(msg.sender == manager);
220     require(_isExcluded(account));
221         for (uint256 i = 0; i < _excluded.length; i++) {
222             if (_excluded[i] == account) {
223                 _excluded[i] = _excluded[_excluded.length - 1];
224                 _excluded.pop();
225                 userbalanceOf[account] = userbalanceOf[account].mul(transferRate);
226                 break;
227             }
228         }
229     }
```

```
253     function updateLPMultiplier(uint256 newLPMultiplier) external {
254         require(msg.sender == manager);
255         require(newLPMultiplier >= 10 && newLPMultiplier <= 200, 'Multiplier is out of the acceptable range!');
256         LPRewardMultiplier = newLPMultiplier;
257     }
258
259     // Function can only be used by the contract owner
260     // It is used to set the reward multiplier for the trading competition reward pool
261     // Initial value is set to 20 (5600 PERA/day)
262     function updateTCMultiplier(uint256 newTCMultiplier) external {
263         require(msg.sender == manager);
264         require(newTCMultiplier >= 10 && newTCMultiplier <= 100, 'Multiplier is out of the acceptable range!');
265         TCRewardMultiplier = newTCMultiplier;
266     }
267
268     // Function can only be used by the contract owner
269     // It is used to set the minimum PERA transaction required for the trading competition
270     // Initial value is set to 100
271     function updateminTCamount(uint256 newminTCamount) external {
272         require(msg.sender == manager);
273         require(newminTCamount >= (10 * 10 ** decimals)  && newminTCamount <= (1000 * 10 ** decimals), 'Amount is out of the acceptable range!');
274         minTCamount = newminTCamount;
275     }
276
277     // Function can only be used by the contract owner
278     // It is used to add the contract address of the LP token
279     function addLPToken(address _addr)  external {
280         require(msg.sender == manager);
281         lpTokenAddress = _addr;
282     }
```

```
645     // Funciton for trading competition winners to claim their rewards
646     function getTCreward(uint _bnum) external {
647         require(_bnum > 0,"min 1 ended TC is required.");
648         require(_bnum.sub(1) < showBnum(), 'At least 1 day is Required!');
649         (uint256 _traderReward, uint256 _traderRewardEligible, uint _winnerIndex, uint256 _rewardEmission, uint256 _rewardFee) =
650         require(_traderRewardEligible > 0, 'No Eligible Reward!');
651         if(_winnerIndex != 404) {
652         isPaid[nMixAddrandSpBlock(msg.sender, _bnum.sub(1))] = true;
653         _mint(msg.sender, _rewardEmission);
654         _transfer(address(this), msg.sender, _traderRewardEligible);
655         }
656     }
```

```
720     // Function for staking LP tokens (min 1 LP token is required)
721     function depositLPtoken(uint256 _amount) external {
722
723         LPUserInfo storage user = userInfo[msg.sender];
724         updateRate(totalStakedLP);
725
726         if (user.userLPamount > 0) {
727             uint256 pendingReward = user.userLPamount.mul(LPRate).div(1e12).sub(user.userReflectedLP);
728             if(pendingReward > 0) {
729                 _transfer(address(this), msg.sender, pendingReward);
730             }
731         }
732         if (_amount > 1 * 10 ** LPTokenDecimals) {
733             totalStakedLP += _amount;
734             user.userLPamount = user.userLPamount.add(_amount);
735             BEP20(lpTokenAddress).transferFrom(msg.sender, address(this), _amount);
736         }
737         user.userReflectedLP = user.userLPamount.mul(LPRate).div(1e12);
738     }
```

```
770     // Function is used to withdraw LP tokens from the PERA smart contract
771     function withdraw(uint256 _amount) external {
772
773         LPUserInfo storage user = userInfo[msg.sender];
774         require(user.userLPamount >= _amount, "withdraw: not good");
775         updateRate(totalStakedLP);
776
777         uint256 pendingReward = user.userLPamount.mul(LPRate).div(1e12).sub(user.userReflectedLP);
778         if(pendingReward > 0) {
779             _transfer(address(this), msg.sender, pendingReward);
780         }
781         if(_amount > 0) {
782             user.userLPamount = user.userLPamount.sub(_amount);
783             totalStakedLP -= _amount;
784             BEP20(lpTokenAddress).transfer(msg.sender,  _amount);
785         }
786         if(totalStakedLP == 0){
787             FeeRewPoolLP = 0;
788         }
789         user.userReflectedLP = user.userLPamount.mul(LPRate).div(1e12);
790     }
```

Halborn-10

Documentation: Consider updating the documentation in GitHub for greater ease when contracts are deployed and tested.

Pera-10

Later this week, suggested documentation will be added to GitHub.

New Contract Owner Functions

We added 3 new contract owner specific functions to set the mint rate for the trading competition rewards and LP token staker rewards and another function to set the minimum required PERA transaction to be eligible for the trading competition. Functions are added to make the contract more scalable in case where there are less/more users than expected or the token price goes up/down more than expected.

```
250     // Function can only be used by the contract owner
251     // It is used to set the reward multiplier for LP token stakers
252     // Initial value is set to 20 (0,5 PERA/block)
253     function updateLPMultiplier(uint256 newLPMultiplier) external {
254         require(msg.sender == manager);
255         require(newLPMultiplier >= 10 && newLPMultiplier <= 200, 'Multiplier is out of the acceptable range!');
256         LPRewardMultiplier = newLPMultiplier;
257     }
258
259     // Function can only be used by the contract owner
260     // It is used to set the reward multiplier for the trading competition reward pool
261     // Initial value is set to 20 (5600 PERA/day)
262     function updateTCMultiplier(uint256 newTCMultiplier) external {
263         require(msg.sender == manager);
264         require(newTCMultiplier >= 10 && newTCMultiplier <= 100, 'Multiplier is out of the acceptable range!');
265         TCRewardMultiplier = newTCMultiplier;
266     }
267
268     // Function can only be used by the contract owner
269     // It is used to set the minimum PERA transaction required for the trading competition
270     // Initial value is set to 100
271     function updateminTCamount(uint256 newminTCamount) external {
272         require(msg.sender == manager);
273         require(newminTCamount >= (10 * 10 ** decimals)  && newminTCamount <= (1000 * 10 ** decimals), 'Amount is out of the acceptable range!');
274         minTCamount = newminTCamount;
275     }
```