Gain Game explanation

Ramil Amirov

November 2023

1 Introduction

This protocol represents a win-win lottery on staking. The basic idea is that users investing in our protocol get a chance to win all the earnings that came from all the user's assets earned.

2 Motivation

How can the world of DeFi benefit from this protocol? Let's start with the fact that our protocol is built according to the rules of DeFi-lego, literally we build it based on other staking protocols. That is, we will benefit those protocols as well, because we will providing them liquidity by bringing them user's assets. So, it turns out that our protocol will be beneficial to everyone: both users and protocols. In addition, this idea can be an alternative to the usual staking, which is not interesting to a large number of users, because it brings very little income.

3 Proof

So, how is this going to work? We need to get as fair a chance to win for each user as possible, because it is obvious that a user who invested 100 SOL in us should have a chance to win 100 times more than a user who invested 1 SOL. What solution do we propose? Let's introduce a system of bonus points that will be given to users who keep money in our protocol for the longest amount of time, so that even users who have a small amount of money have a chance to win for believing in our protocol. So, let's move on to the more technical part. First, let's give out chances to win to users (which is logical) and every second we will give out to all tokens from *totalSupply* a chance to win for that second. That is, initially we have a 100 percent chance to win, which we need to distribute among all users and somehow stretch over time. That is: the formula for distributing every second will be as follows:

$$chancePerSecond = \frac{100\%}{ONE\_WEEK\_IN\_SECS \times currentTotalSupply}$$

It turns out, now we've learned to calculate how much each user's chance will increase for each second, like this:

$$chanceGained[user] = balance[user] \times chancePerSecond$$

But I think it's obvious to everyone that it's very expensive to count the odds for users this way, we'd have to go through all users every second and add their odds, which is unbelievably gas-inefficient. Let's change our approach a bit, we'll add a chance per token rather than per user. We can simply calculate by how much the chance of winning for each token increases in the interval where the pool does not change. So: we will update the "chance per token" every time the pool is updated, that is, we will call this function each time any user call any mutative function:

$$chancePerToken += chancePerSecond \times (currentTime - lastMutativeTime)$$

$$lastMutativeTime \rightarrow \text{time of last pool changing}$$

Also to consider that many users will start using our protocols far from the very beginning of the giveaway, so for each user we will memorize the odds on the tokens that have already been played before that user arrives. Also, we should not forget that the same user can withdraw a certain amount of money that he has already deposited or deposit some more. So, to calculate what's the chance of a given user:

$$chance[user] = balance[user] \times (chancePerToken - userStartChance[user])$$

$$userStartChance \rightarrow \text{chance per token before this user action}$$

Winner determination will be on-chain and will be implemented using VRF.