Hardcore Bank
Implementing Time Deposits with Smart Contracts to Achieve Goals by Conditioning with Negative Reinforcers

kekeho [†]

January 20, 2022

Abstract

People save money to achieve some goal. However, some people with weak willpower are often tempted by the temptation in front of them and give up saving halfway. I devised a method to help people save money by conditioning them with negative reinforcers to solve this problem. I implemented a time deposit account with an Ethereum smart contract, in which assets decrease unless a fixed amount is deposited every month.

1 Background

People often take action of saving. For example, they save money to go abroad as an extended vacation or to save money for an emergency. However, saving is a constant struggle against the temptations in front of us. It is difficult for weak-willed people to overcome the frequent desire to splurge and achieve the target amount.

Therefore, I think that I can help users save money by conditioning them appropriately.

2 Reinforcement

In instrumental conditioning in behavioral psychology, due to the environmental phenomenon that follows the action, it is said that behavioral occurrence probability increases or decreases.

Reinforcement is a process in which the behavioral occurrence probability increases with accompanying stimulus. There are positive and negative concomitants between behavior and reinforcement.

There are two types of reinforcement, positive and negative. Positive reinforcement causes pleasant or desirable stimulus after behavioral response, and it increases the frequency of the behavioral response. Negative reinforcement removes unpleasant or disgusting stimulus after behavioral response, and makes it occur frequently of the behavioral response[1].

3 Applying Negative Reinforcement to Saving Money

I suppose that the negative reinforcement being introduced in Section 2 could contribute to achieving the target amount in savings.

I hypothesize that negative reinforcement could be achieved by happening a risk that the saved money will be lost if the user does not take action to deposit money into the piggy bank.

Based on this theory, I devised a time deposit account as follows. First, opening a time deposit account being set target amount and monthly deposit amount. Users can withdraw this account only when the remainder exceeds the target amount.If there is a month when the transfer amount is less than the set deposit amount, 20% of the remainder will be confiscated each time.

Users will be exposed to the fear of losing money they have saved. Therefore, I expect the user will take action to transfer money as planned every month in order to avoid losing the remainder. Due to this kind of negative reinforcement, I helps users save money constantly.

4 Implementation

I implement the time deposit account shown in Section 3 with an Ethereum smart contract.

4.1 ERC777 Token

The currency to be deposited is the ERC777 token[2]. ERC777 is a standard interface to represent Fungible Token in smart contracts.

As the preceding Fungible token standard, ERC20[3] has already existed. Mostly ERC20 has the following functions.

- Transfer a token from an account to another.

- Get information of the current token remainder of an account.

- Obtain the total supply of tokens on the network.

- Approve whether the amount of tokens held by an account can be used by a third-party account.

[†]Keio University, Faculty of Environment and Information Studies B1
e-mail: kekeho@sfc.wide.ad.jp
Real name: Hiroki TAKEMURA
KG: Bcali

ERC777 is an extension of ERC20. It is backward compatible with ERC20. It is not only sends and receives tokens, but also defines the entire life cycle of tokens such as Mint and Burn.Another feature is that the Hook function for sending and receiving tokens is defined.

In this contract, I adopted the ERC777 token with Hook function because it is necessary to perform the process of storing the history when the token is received.

4.2 System

The entire system is shown in Figure 1.

4.2.1 Contract

In this section, I describe the implementation of the Hardcore Bank contract.

This contract is deployed only once on the Ethereum network by the bank owner. The contract record account information showing the form shown in Code 1, with the following fields. This allows the system to know the account information and which account holds how much of the various ERC777 tokens which the contract holds. In addition, the current account remainder is dynamically calculated by `recvList`.

The contract implements the methods necessary to receive ERC777 tokens, so that the user can make a deposit. When a user transfers tokens to the contract, they can specify the account ID in the data section and transfer the tokens to a specific account in the contract.

Code 1: Basic Struct

```
1  pragma solidity ^0.8.0;
2
3  struct Config { // Information of account
4      uint256 id;
5      address owner;
6      string subject;
7      string description;
8
9      address tokenContractAddress;
10     uint256 targetAmount;
11     uint256 monthlyRemittrance;
12
13     uint256 created;
14
15     bool disabled;
16 }
17
18 struct RecvTransaction {
19     address from;
20     uint256 amount;
21     uint256 timestamp;
22 }
23
24 Config[] private accountList;
25 mapping(uint256 => RecvTransaction[])
       private recvList; // Account ID =>
       RecvTransaction[]
```

4.2.2 Front-end

In this section I introduce the web interface that manipulate the contract introduced in Section 4.2.1. The implemented functions are as follows.

- Opening an account

- Checking the balance of the account

- Withdrawal of funds

This web interface is intended to be operated from a browser equipped with a Metamask compatible wallet. The frontend implemented in Elm can access the Ethereum blockchain via web3js and the Metamask wallet. When a user operates, the function of the Hardcore Bank contract can be called via web3js, and the result of the operation can be reflected on the blockchain.
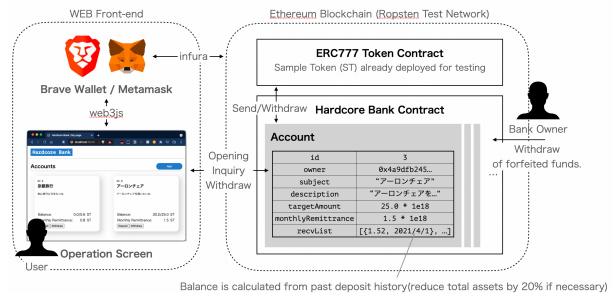

Figure 1: Diagram of the overall system

5 Discussion

In this study, by using negative reinforcement in instrumental conditioning, we implemented a time deposit that helps the user save money.

In the future, we need to conduct empirical experiments to verify whether this time deposit can help the user. In addition, we tentatively set the rate of decrease in the balance to 20% when the user does not send money, but we need to find an appropriate value for this rate.

In addition, as of January 2022, the cost of gas for ERC777 Token remittance is rising, which means that the monthly remittance will cost much money. Because of this, it is not an economical way to save money at this stage. Thus, it is necessary to conduct further research to reduce the cost of gas.

References

[1] S. Nolen-Hoeksema, B. L. Fredrickson, G. R. Loftus, and C. Lutz, *Atkinson & Hilgard's Introduction to Psychology (Japanese Edition)*, 16th ed. Kongo shuppan, 2015, ISBN: 9784472414388.

[2] J. Dafflon, J. Baylina, and T. Shababi, *EIP-777: Token Standard*, https://eips.ethereum.org/EIPS/eip-777, (Accessed on 01/20/2022).

[3] F. Vogelsteller and V. Buterin, *EIP-20: Token Standard*, https://eips.ethereum.org/EIPS/eip-20, (Accessed on 01/20/2022).