Susu Smart Contract Design

Definitions

Round - The time during which each participant in the Susu pays the contribution amount and one participant receives the pool of money.

Cycle - The time during which each participant receives money once. Equal to number of participants * round.

Susu Lifecycle

1. **Starting a group:** A Susu group is created by deploying a new Susu contract on the blockchain. The deployer of the contract is automatically designated as the owner and they are added (by default) to their own group. They are the group organizer and get to initialize the group with the properties listed below. Note that once the group is created all items are unchangeable. I also added variable names to be used in the rest of this document for brevity.
    a. Name of the group [*groupName*]
    b. Number of people in the group [*groupSize*]
    c. The frequency (monthly? weekly?) [*frequency*]
    d. The contribution rates (1 ETH, 0.1 ETH) [*contribAmtWei*]
2. **Joining:** Anyone can join the group. To join the group you must call a certain method on the contract. The new account's address then gets recorded in the Susu contract.
3. **Contributing:** Once the group is full (the number of those who have Joined from #2 matches "number_recipients") then it is ready for contributions. Each recipient visits the website and clicks the "contribute" button to chip in their share. The contribution amount MUST match the contribution_amt or it will be rejected and the contribution will not count. The page reflects who has and has-not contributed for this cycle and how much each person contributed. The contract records how much each recipient has contributed.
4. **Paying Out:** The group owner is then responsible for clicking the "Pay Out" button at the end of the cycle. The owner cannot invoke this action if the group is not full of people and all people have posted their contributions for this cycle.
5. **Leaving a group:** Anyone in the group can leave anytime by clicking the "I'm a little bitch" button. When they do this they get their deposit back along with any contributions they've made during the *current* cycle.

6. **Terminating a group:** Only the group owner can finally terminate a group. Any remaining members will be given back their deposits and any contributions that they have made to the current cycle.

Other Supporting Elements

1. We can potentially use components from OpenZeppelin to help make a secure contract.
    a. Ownership (Example in CryptoZombies)
    b. Math
    c. Payment
2. Create a normal legal contract version of the Susu terms and integrate with OpenLaw?
3. Think about potentially adding SageWise.

Other Notes

- Online Solidity IDE/Compiler
- For securities look at using ERC-884.
- Etherscan: For checking the status of items on blockchains?
- Also worth reviewing Consensys' "Smart Contract Best Practices" as we build this out.
- Deploying from the browser article HERE