Ethan Winters
CMSC389C
Final Project Report

What the contract does

This contract is a gambling contract that manages a simple game between up to ten participants (for the sake of keeping the risk relatively low and the reward relatively high). If you want to enter the game, you stake 1eth and make a guess at a number 1-100 (inclusive). Once the game is full, or the game host decides to close betting, a random number is generated between 1-100 and the winner[s] are chosen based on closeness to the random number. If more than one person is the same proximity away from the number, they split the entire pot, but the only people who will win are those closest to the number in the first iteration.

How the contract should run

When running the contract, you must deploy it using an account. That account cannot bet in the game. Once you switch to an account that can play in the game (has 1eth, isn't the host) then you can enter 1 in the value box and put any number between 1-100 in the joinGame function. That user will then be entered into the game, and not allowed to bet again. Once you want to close the game, select generateRandomNumber, which triggers a random number to be generated based on block hash and will payout the pot fairly to the winner[s].

How to call functions/expected behaviors

joinGame and generateRandomNumber work as described above. Besides those, there are some getter methods for testing purposes. getPlayers gets all of the players' addresses getPot returns the current value of the pot (10,000*numberPlayers), getWinnersAddresses returns an array of winners' addresses, and getWinningNumber returns the final winning random number.

Constraints

1. *Variable participants*: the contract accepts up to 10 players, but can be run with as little as two. I can set the max amount to as high as I want, but for the sake of keeping the game somewhat winnable, 10 seems about right.
2. *One payable function*: the chooseWinner function in the Solidity file transfers eth to the winner[s] of the game.
3. *Punish dishonest participants*: the contract does not let users enter the game more than once, and you cannot stake more than 1eth.
4. D*isallow new participants after the contract is over/finished*: the contract does not let new users place bets once the generateRandomNumber function has been called, since that could lead to unfair advantages.