University of Edinburgh	*Fall 2024*
Blockchains & Distributed Ledgers	

Smart Contract Programming Coursework Assignment

(Total points = 100, BDL mark weight = 30%)

Due: Wednesday 7.11.2024, 12.00 (noon)

Please remember the good scholarly practice requirements of the University regarding work for credit. You can find guidance at the School page: https://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct. This also has links to the relevant University pages.

In this assignment, you will write your own smart contract.

The smart contract should emulate the following game between two players, A and B, that proceeds in rounds. The first player A picks a number in [1,100], say denoted by nA. The second player does the same picking the number nB. Subsequently players A and B submit their guesses for each other's value denoted by gA and gB respectively. The player who wins is the one with the smallest value |gB – nA|, |gA - nB|. The reward is nA+nB. If it so happens that |gB – nA| = |gA - nB| then the game is a draw and no player wins (or loses) any money.

Example. Two players, A and B, each with 1,000,000,000 wei in their wallets, start a game. A picks nA = 40 and B picks nB = 34. Then, A guesses gA = 84 for B's number and B guesses gB = 50. B's guess is closer and hence he wins. After the game ends, B's balance is 1,000,000,074 wei (possibly minus some gas fees, if necessary).

After a game ends, any two players should be able to start a new game on the same contract.

You should implement the smart contract and deploy it on Ethereum's testnet, Sepolia. Your contract should be as *secure*, *gas efficient*, and *fair* as possible. Specifically, your contract should: i) implement the game that is described above; ii) not allow one of the two players to cheat (in other words, you should prevent as many attacks as reasonably possible). After you have guaranteed these two facts, you should try to make it as efficient and/or fair as possible and detail the tradeoff choices you made.

Commented [CO1]: It might be useful to clarify that gB is the guess made by player A

Commented [CO2]: It is not specified how the reward is calculated (and it's not too clear from the example either)

After deploying your contract, you should engage with at least one other student and play a game on their contract; you may use Piazza to find a partner. Before you engage with a fellow student's smart contract, you should evaluate their code and analyze its features in terms of *security, efficiency,* and *fairness* (cf. Lectures 3-4).

Submission

You should submit **two files** via Learn (in the same Learn submission) choosing the option "Submit via Gradescope." Marking is done anonymously, so read the submission instructions on Learn carefully and *do not* include your name or student number in any of the submitted files (or the name/number of the fellow student with whom you interacted).

Commented [CO3]: Just to confirm, can we still use gradescope when they submit via Learn? is there anything we need to do for it?

First, a solidity file that contains the code of your smart contract. The name of the file should be your exam number (e.g., *B123456.sol*).

Second, a PDF report that contains:

- A detailed description of the high-level decisions you made for the design of your contract, including (but not limited to):
    - Who pays for the reward of the winner?
    - How is the reward sent to the winner?
    - How is it guaranteed that a player cannot cheat?
    - What data type/structures did you use and why?
- A detailed gas evaluation of your implementation, including (but not limited to):
    - The cost of deploying and interacting with your contract.
    - Whether your contract is fair to both players, including whether one player has to pay more gas than the other and why.
    - Techniques to make your contract more cost efficient and/or fair.
- A thorough list of potential hazards and vulnerabilities that *may* occur in the contract. Provide a detailed analysis of the security mechanisms you use to mitigate such hazards.
- A detailed description of the tradeoffs and choices you made, e.g., between security and performance, fairness and efficiency, etc.
- Your analysis of your fellow student's contract (along with relative code snippets of their contract, where needed for readability), including (but not limited to):
    - Is their implementation more secure/efficient/fair than yours?
    - Any vulnerabilities discovered?
    - How could a player exploit these vulnerabilities to win a game?
- The address of your contract on Sepolia.
- The code of your contract. *(Note: The contract should be both at the end of your PDF report and submitted as a separate file, as described above.)*

The PDF report, excluding the contract's code, should be at most 10 pages (font size at least 11, margin at least 1 inch all around). The name of the file should be your exam number (e.g., *B123456.pdf*).

Marking details

Marking follows the Common Marking Scheme and the intention is that each submission's mark will reflect in which grade of the scheme the submission lies. The marks will be roughly distributed as follows.

70% of the marks will be allocated regarding security. This includes the description in your report, the security of your contract, and the analysis of your fellow student's implementation. Security will be evaluated with respect to attacks discussed in Lectures 3-4 and, if the contract is susceptible to some of them, marks will be deducted appropriately.

30% of the marks will be allocated regarding gas efficiency and fairness. This again includes your report's text, your code, and your analysis of the other student's contract.