| **University of Edinburgh** | *Fall 2022-23* |
|---|---|
| **Blockchains & Distributed Ledgers** | |

# Assignment #2 (Total points = 100)

## Due: Monday 31.10.2022, 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

### Smart Contract Programming Part I: Rolling the dice

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

The smart contract should emulate a dice roll. In other words, the contract should enable the computation of a *random* number *n* in the range [1, 6]. If *n* is 1, 2, or 3, then A wins; otherwise, B wins. If A wins, A is rewarded *n ETH*; if B wins, B is rewarded *(n - 3) ETH*. After a game ends, two new players should be able to start a new game on the same contract.

**Example.** Two players, A and B, each with 100 ETH in their wallets, start a game. The produced number is 5, so B wins. After the game ends, B's balance is 102 ETH (minus some gas fees, perhaps, if necessary).

You should implement the smart contract and deploy it on the course's Ethereum testnet. Your contract should be as *secure*, *gas efficient*, and *fair* as possible. 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).

First, a solidity file that contains the code of your smart contract. The name of the file should be your student number (e.g., *s1000000.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):
    - Any vulnerabilities discovered?
    - How could a player exploit these vulnerabilities to win a game?
- The transaction history of an execution of a game on your contract.
- 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 transaction history and the contract's code, should be at most 10 pages (font size at least 11, margin at least 1 inch all around).