University of Edinburgh	*Fall 2021-22*
Blockchains & Distributed Ledgers	

Assignment #2 (Total points = 100)

Due: Monday 1.11.2021, 16.30

Smart Contract Programming Part I: Matching Pennies

This assignment will focus on writing your own smart contract to implement the Matching Pennies game. The contract should allow two players (A, B) to play a game of Matching Pennies at any point in time. Each player picks a value of two options (for example, the options might be {0, 1}, {'a', 'b'}, {True, False}, etc). If both players pick the same value, the first player wins; if players pick different values, the second player wins. The winner gets 1 ETH as reward. After a game ends, two different players should be able to use your contract to play a new game.

Example. Let two players A, B, each with 100 ETH, who play a game. A picks 0 and B picks 0, so A wins. After the game ends, A's balance is 101 ETH (perhaps minus some gas fees, 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 other students' contracts 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 and fairness (refer to Lecture 04).

You should submit 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/structure did you use for the pick options 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)
- A description of 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 the game?
- The transaction history of an execution of a game on your contract.
- The code of your contract.