

## Assignment 3

Due: 11:55 pm 29 Aug 2022

Total Mark: 100 (17% of Final Mark)

General Instructions: Please read the following instructions carefully.

**Q1.** Create a simple blockchain system to enable two parties, Alice and Bob, to play a tic-tac-toe game.

Both players have an initial block to start their play. Let Alice always start first for simplicity. The initial block, *block0.json*, for both Alice and Bob is set as follow:

```
{
  "TxID": 0,
  "Hash": "This is the genesis block.",
  "Nonce": 0,
  "Transaction": []
}
```

*Figure 1. Block0.json*

At each turn, a player (e.g., Alice) who takes the turn creates a new block in JSON format and send it to the other player (e.g., Bob) who plays tic-tac-toe together. Each block is formed as follows:

```
{
  "TxID": <TxID>,
  "Hash": <Hash>,
  "Nonce": <Nonce>,
  "Transaction": <Transaction>
}
```

The entities in the above block are computed in a way described as follows:

1. "TxID" is an incremental counter, which increased by 1 on each turn.

$$\text{<TxID> of a new block} = \text{<TxID> of the last block} + 1$$

2. "Hash" is the output of the SHA256 hash of the last block.
3. "Nonce": the player increases the value of "Nonce" from 0 by 1 until the value of the hash output of the current block is less than  $2^{244}$ .

$$\text{SHA256}(\text{<The current block>}) < 2^{244}$$

4. "Transaction" is the choice of the player in its turn. It contains the player name and its choice of the space on the tic-tac-toe board (e.g., ["Alice",B]). To make your code simple, assume that Alice and Bob take a random empty space in their turn.

**For example,** Assume that we use the following grid to play a tic-tac-toe game.

A	B	C
D	E	F
G	H	I

The first block (block1.json) is generated by Alice as follows.

```
{
  "TxID": 1,
  "Hash": "...",
  "Nonce": 11,
  "Transaction": ["Alice",B]
}
```

Note that in your program, Alice must choose a space, randomly. Then, Alice saves the block in her storage and sends it to Bob.

Once Bob receives the block, Bob validates the received block and save it to its storage. Then, Bob creates a following block and send it to Alice:

```
{
  "TxID": 2,
  "Hash": "...",
  "Nonce": 742,
  "Transaction": ["Bob",A]
}
```

Note that in your program, Bob must randomly choose a space among the spaces not taken previously.

The game will continue **until all spaces in the board are taken**. Alice and Bob keep each block in a separate file (block0.json, block1.json, ..., block9.json) in their local storage. Before each player executes their turn, it must check that the block received is valid.

Make the players exchange blocks online. You can use the **PubNub package** to implement this in Python.



### **Marks**

- Block generation process implementation. (30 marks)
- Block validation process implementation. (10 marks)
- The other basic requirements of the system (e.g., implementation of a tic-tac-toe game and handling JSON format.) (30 marks)
- Implementation of blocks exchange protocol (20 marks)
- Report (10 marks)

### **Submission**

Make a folder named `Assignment3` and include

- Programs for Alice and Bob.
- The history of one of the completed games (e.g., `block0.json`, `block1.json`, ..., and `block9.json`)
- Report explaining the requirements to execute your program and the expected outcome of your program.

Compress the `Assignment3` folder using a zip program to create `yourSurname_Assignment3.zip`.

Use Moodle to upload your zip file.