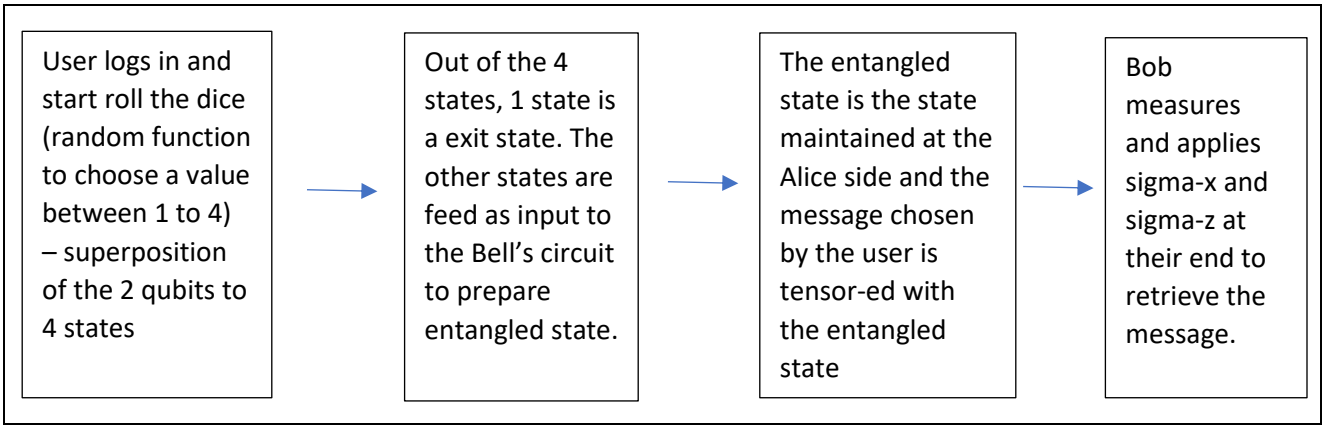
QHack Challenge 2022 – IBM challenge

Objective:

Part A: to create a simple game using QC circuits as a base to show how QC can be used in the game development.
Part B: to develop user interface to integrate the QC function developed in Part A to be used in a enhanced way.

Our Team theme:
We planned to design a simple game using mainly the superposition and the quantum parallelism in mind. So, we thought of our childhood game Ludo where we roll the dice and get a value and based on the value, we play movement games or point collection games. So, based on this idea we are developing a simple game so that the user is lucky enough to get the success state always and able to transmit information to the other end using Superdense Coding to transfer classical data over a quantum channel. If the Quantum computer want to stop the user from playing, then it generates the state for which exit state is assigned higher probability.



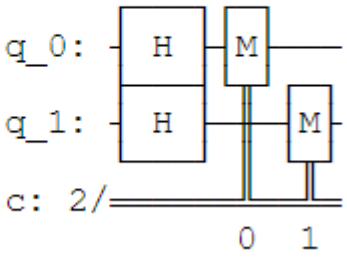
So, we designed the game to be enhanced later in a way to collect points or to play continuously until the user encounters the exit state. The exit state is randomly assigned by system in each round of play.

Technical design:

Step 1: User logs in and rolls the dice. The random generator is 2 qubits put in superposition by applying Hadamard gates and then applying the measurement to get the random probabilistic value every time. Out of the 4 states, 3 states allows to proceed to next stage, while 1 state is the exit state that stops the game.

```
In [87]: circuit.draw()
```

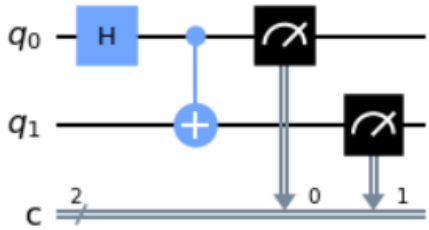
Out[87]:



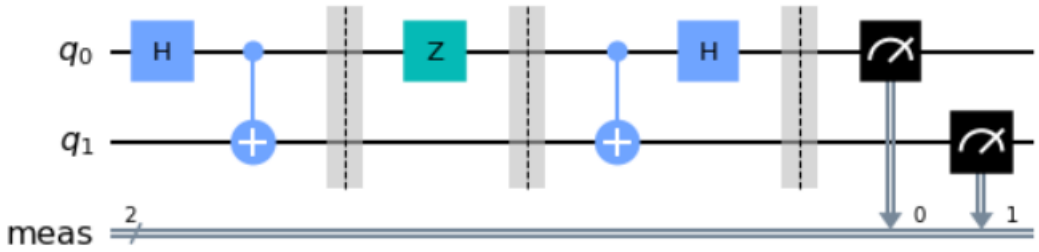
Step 2: The output of the step1 is a random state that is given as input to the Bell's state to produce the entangled states. So out of the 4 state, the Bell circuit is given only 3 states randomly at each round of game and then the entangled states are generated.

```
[7]: qc.draw()
```

[7]:



Step 3: So based on the entangled state, Alice and bob maintain the same state and we send a information at Alice end so that with the changes measured at Bob's end. Bob applies sigma-x and sigma-z to decode the information.



Results:

1) We are able to successfully make the user utilise the superposition and measurement to use as random number generator like the dice usage.
2) With the random state achieved, we are able to generate entangled states successfully.
3) Then we are able to use the entangled state to teleport the message to the other end.