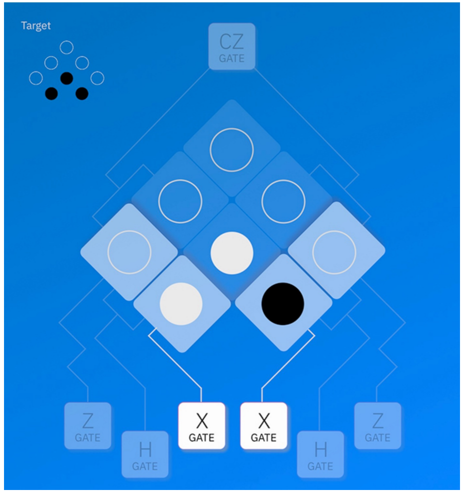
Qiskit Textbook
§9 9.1 Hello Qiskit Game

Emi ADACHI

Agenda

1. Hello Quantumのアプリの紹介
2. スマホにHello Quantumゲームをインストールする方法
3. Hello Qiskit Gameの設定とルール
4. ゲートの種類
5. 量子プログラミングを作ってみましょう
6. 古典ビットと量子ビットの違いと特徴
7. 参考文献、引用文献

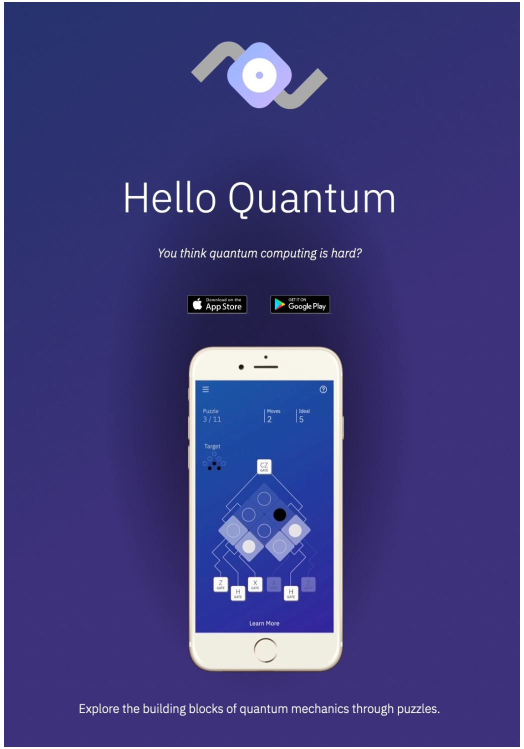
1. Hello Quantumゲームの紹介



量子コンピューターの仕組みをゲームで実感することができます。スマートフォンやタブレットからでも使うことができます。

https://www.ibm.com/blogs/research/2018/07/hello-quantum/

2.Hello Quantumゲームのインストール方法



出典：http://ibm.biz/helloq19

1. 1 Hello Qiskit Gameの設定

OFFの状態$|0\rangle$, $|+\rangle$

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$|+\rangle = |0\rangle + |1\rangle$$

ONの状態$|1\rangle$, $|-\rangle$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$|-\rangle = |0\rangle - |1\rangle$$

重ね合わせの状態

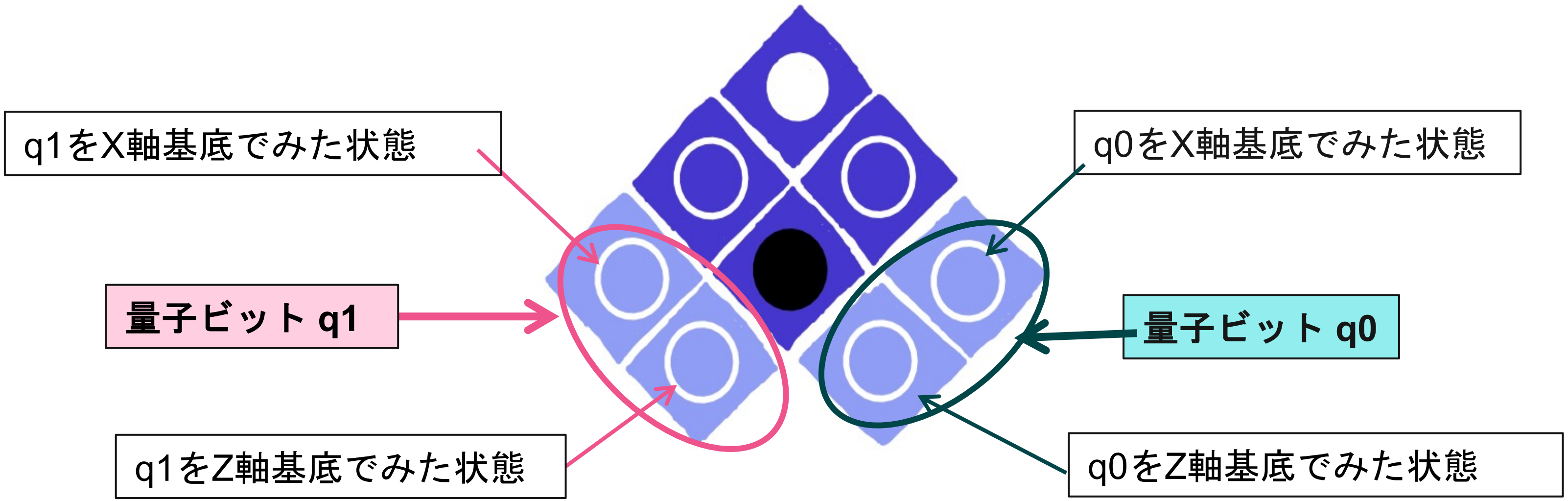
ONとOFFのどちらかの
状態

出典：http://ibm.biz/helloq19

1.2.1 ゲームのルール1

下側の２対の円は、それぞれの1量子ビットを表す。

つまり、このゲームは、2量子ビットの状態を操作するゲームである。

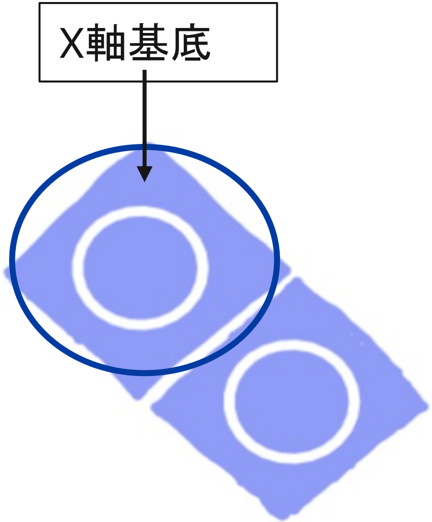


出典：http://ibm.biz/helloq19

1.2.2 ゲームのルール2

量子ビットの上側の円：

X軸基底の方向で見たときの$|+\rangle$と$|-\rangle$をそれぞれOFFとONにみなしたものである。



OFF: $|+\rangle$

$|+\rangle = |0\rangle + |1\rangle$

ON: $|-\rangle$

$|-\rangle = |0\rangle - |1\rangle$

重ね合わせ

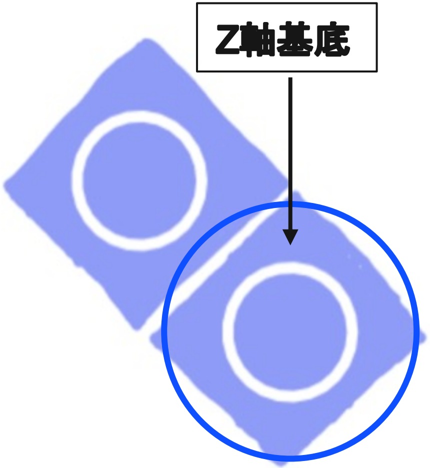
ONとOFFのどちらかの状態

出典：http://ibm.biz/helloq19

1.2.2 ゲームのルール３

量子ビットの下側の円：

Z軸基底の方向で見たときの$|0\rangle$と$|1\rangle$ をそれぞれOFFとONにみなしたものである。



OFF: $|0\rangle$

ON: $|1\rangle$

重ね合わせ

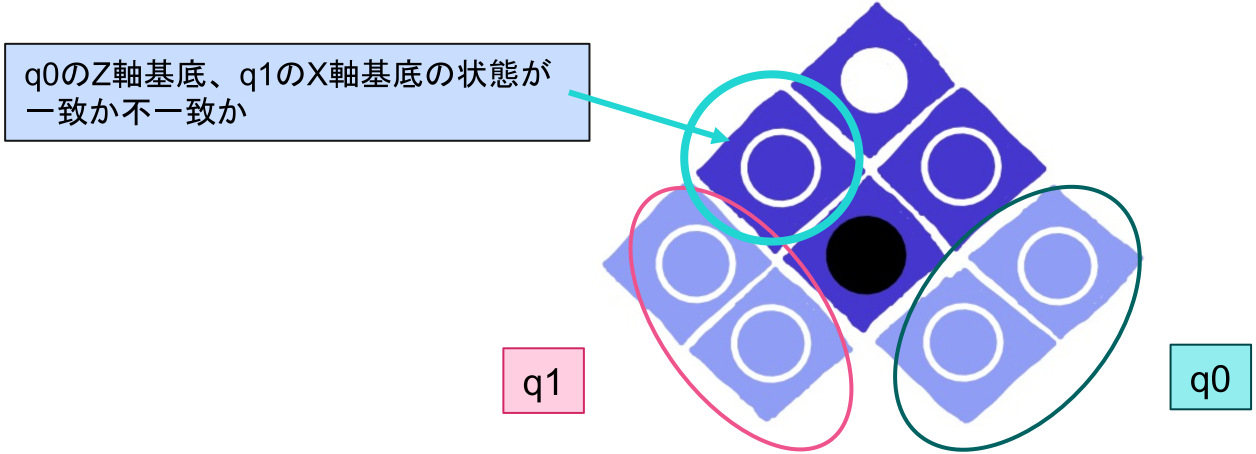
ONとOFFのどちらかの状態

出典：http://ibm.biz/helloq19

1.2.3 ゲームのルール4

上側の４つの円の色:

２つの量子ビットをZ軸基底またはX軸基底で測定した値が一致しているかどうかを表現

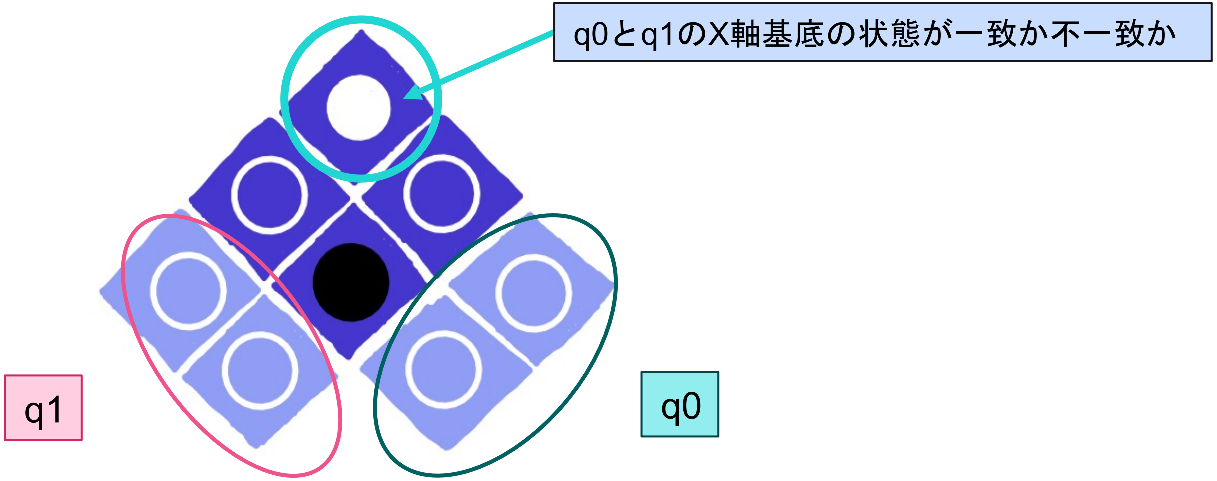


出典：http://ibm.biz/helloq19

1.2.3 ゲームのルール5

上側の４つの円の色:

２つの量子ビットをZ軸基底またはX軸基底で測定した値が一致しているかどうかを表現

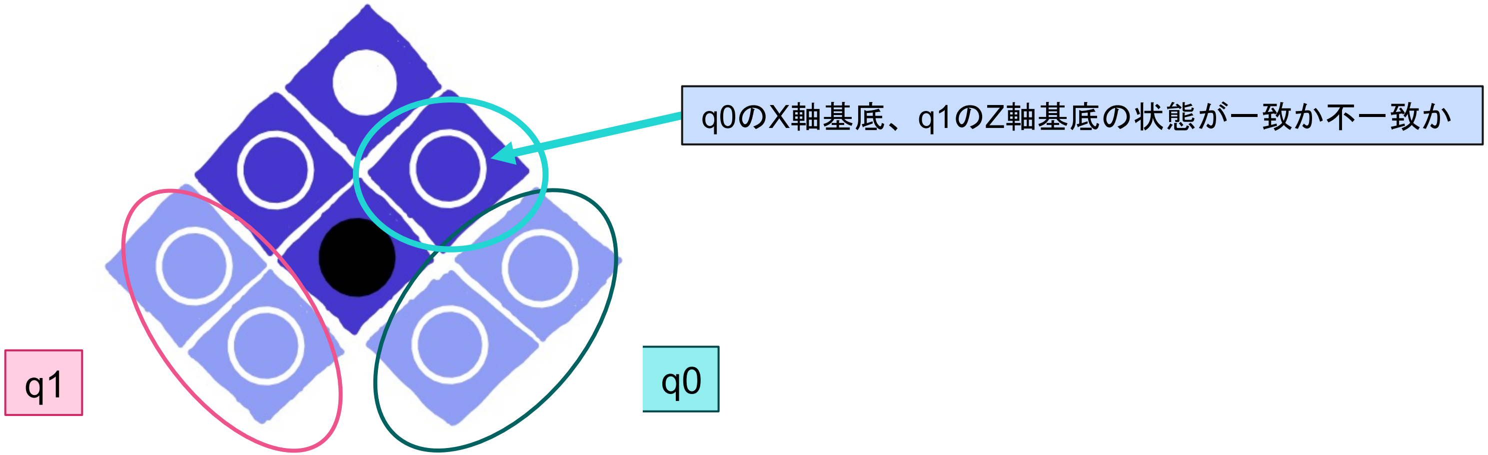


出典：http://ibm.biz/helloq19

1.2.3 ゲームのルール6

上側の４つの円の色:

２つの量子ビットをZ軸基底またはX軸基底で測定した値が一致しているかどうかを表現

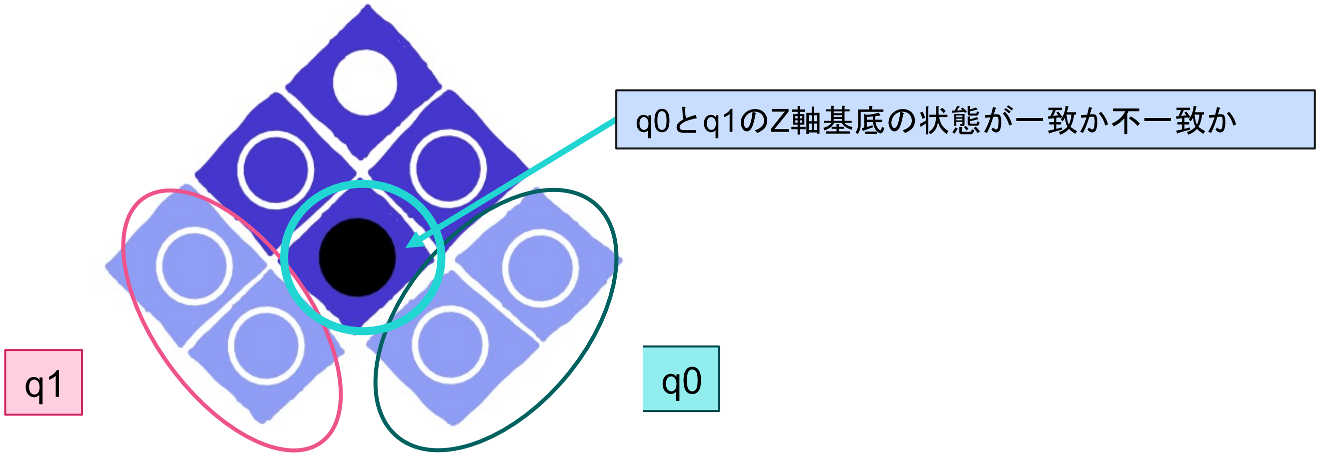


出典：http://ibm.biz/helloq19

1.2.3 ゲームのルール7

上側の４つの円の色:

２つの量子ビットをZ軸基底またはX軸基底で測定した値が一致しているかどうかを表現

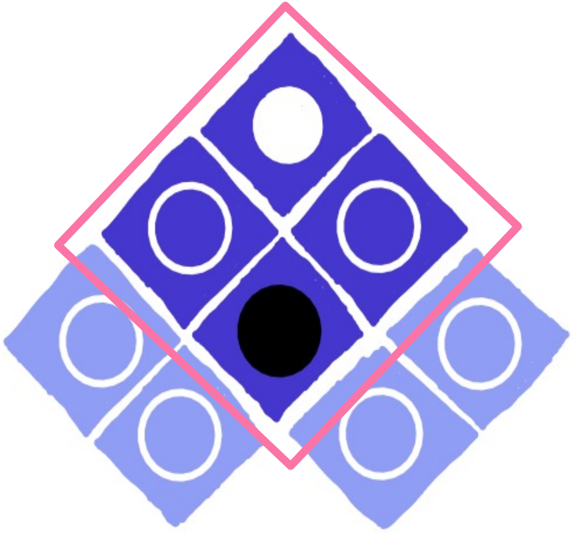


出典：http://ibm.biz/helloq19

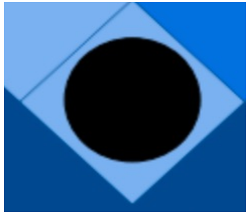
1.2.3 ゲームのルール8

上側の４つの円の色:

２つの量子ビットをZ軸基底またはX軸基底で測定した値が一致しているかどうかを表現



一致の状態



00または11または
00と11

不一致の状態

01または10
または01と10

重ね合わせの状態

01,10,10と11が
混じった状態

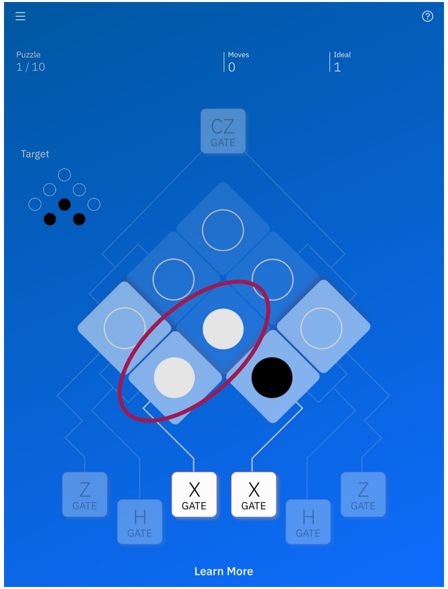
出典：http://ibm.biz/helloq19

2. 量子ゲートの種類

1. Xゲート(NOTゲート)
2. H（アダマール）ゲート
3. Zゲート
4. CZゲート(CNOTゲート)

2.1.1 Xゲート

Xゲートに繋がっている円の白と黒を反転させる操作



Xゲートがつながっている円:
Z軸基底の方向で見たときの $|0\rangle$ と $|1\rangle$ をそれぞれ黒(OFF) と白(ON) にみなしたものである。
Xゲートは、古典ゲートのNOTと同じで、OFF$|0\rangle$とON$|1\rangle$を反転させる操作である。

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

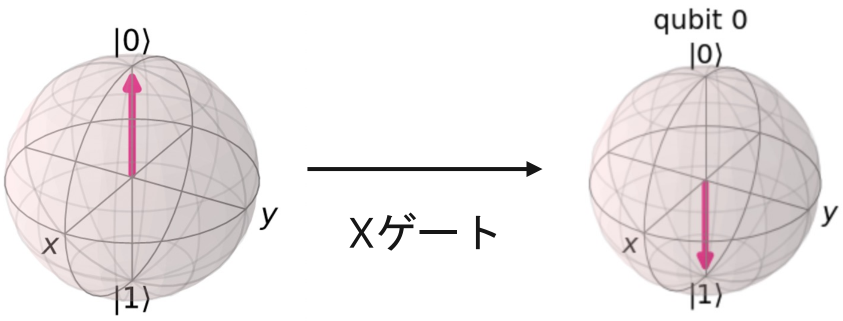
これを重ね合わせ状態に作用させても、結果が変わらない。

$$X\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(X|0\rangle + X|1\rangle) = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)$$

出典：http://ibm.biz/helloq19

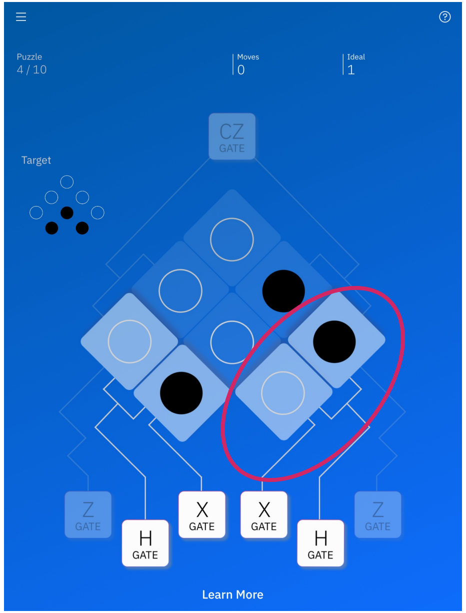
2.1.2 Xゲートの数理的背景

in	out
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$



2.2.1 Hゲート

Xゲートにつながった円とZゲートにつながった円を入れ替える操作



Zゲートにつながっている円は、X軸基底の方向で見たときの、
$|+\rangle$ と $|-\rangle$ をそれぞれOFFとONにみなしたものである。
$|0\rangle$ と $|1\rangle$ 、 $|+\rangle$ と $|-\rangle$ はHゲートを通じて次のような関係で繋がっている。

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

逆もまた然りで、|+>と|->はHを使って、|0>と|1>に戻る。

$$H|+\rangle = \frac{H}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{H}{\sqrt{2}}|0\rangle + \frac{H}{\sqrt{2}}|1\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |0\rangle - |1\rangle) = |0\rangle$$

$$H|-\rangle = \frac{H}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{H}{\sqrt{2}}|0\rangle - \frac{H}{\sqrt{2}}|1\rangle = \frac{1}{2}(|0\rangle + |1\rangle - |0\rangle + |1\rangle) = |1\rangle$$

Hゲートは**X軸基底での見方とZ軸基底の見方を入れ替える操作である。**

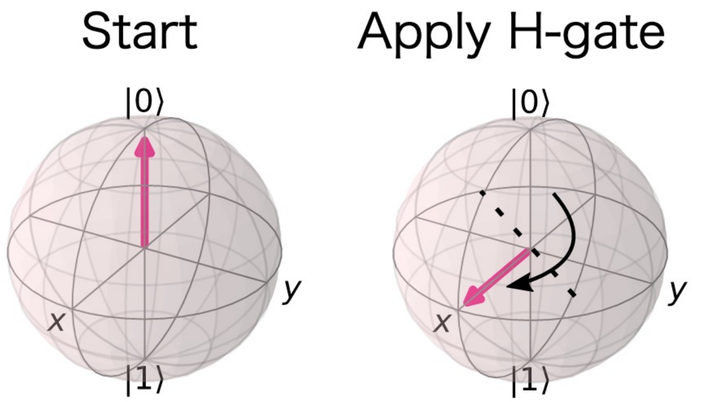
出典：http://ibm.biz/helloq19

2.2.2 Hゲート（アダマールゲート）

重ね合わせ状態を作るときに使うゲートで、45度中心に180度回転させる操作。

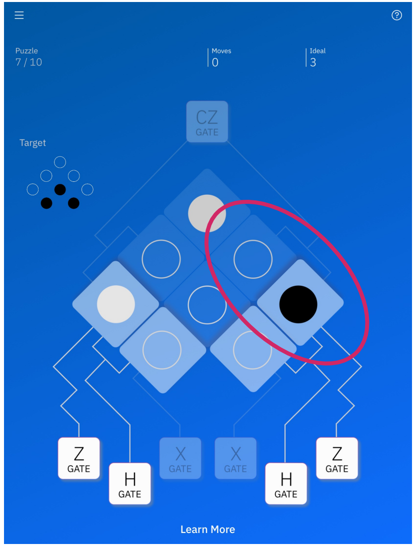
Hゲートを２回作用させると状態は変化しない。

in	out
$\|0\rangle$	$\frac{\|0\rangle + \|1\rangle}{\sqrt{2}}$
$\|1\rangle$	$\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}$



2.3.1 Zゲート

Zゲートにつながっている円の白と黒を反転させる操作



Zゲートは、 $|0\rangle$に作用させても、そのままである。
$|1\rangle$の符号は変える操作をします。

$$Z|0\rangle = |0\rangle$$
$$Z|1\rangle = -|1\rangle$$

それを踏まえて、|+>と|->にZゲートを作用させると、

$$Z|+\rangle = \frac{Z}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(Z|0\rangle + Z|1\rangle) = \frac{1}{2}(|0\rangle - |1\rangle) = |-\rangle$$
$$Z|-\rangle = \frac{Z}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(Z|0\rangle - Z|1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle) = |+\rangle$$

Zゲートとは、X軸基底で見たときに、ONとOFFを入れ替えるという操作に相当。
ゲーム上での動きと同じである。

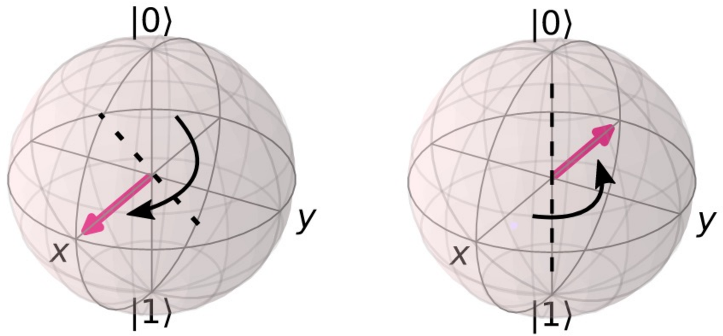
出典：http://ibm.biz/helloq19

2.3.2 Zゲート

パウリの位相フリップゲート

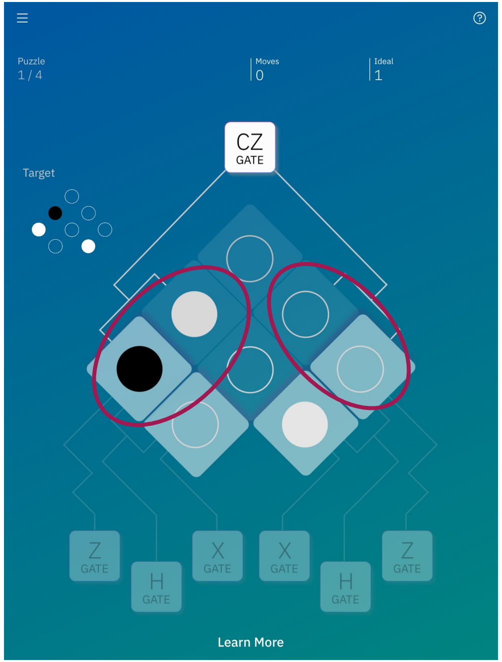
ブロッホ球でZ軸方向に回転するもの

in	out
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$-\|1\rangle$



2.4.1 CZゲート

CZゲートにつながった円をそれぞれ入れ替える操作



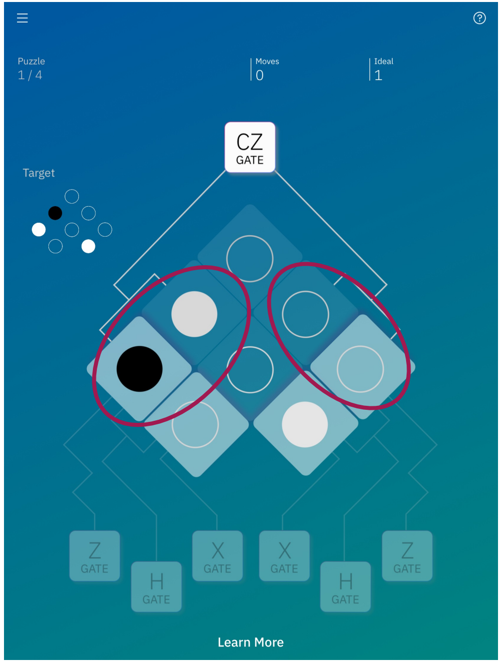
CZゲートは、コントロールビットが|0>であれば、ターゲットビットの状態は変えず、|1>であればターゲットビットの状態にZを作用させるものである。
１つ目の量子ビットをターゲットビット、２つ目の量子ビットをコントロールビットとすると、以下のような数式で表現される。
識別のため、添え字にターゲットビットはtを、コントロールビットにはcをつける。

$$CZ(|0\rangle_t \otimes |0\rangle_c) = |0\rangle_t \otimes |0\rangle_c$$
$$CZ(|1\rangle_t \otimes |0\rangle_c) = |1\rangle_t \otimes |0\rangle_c$$
$$CZ(|0\rangle_t \otimes |1\rangle_c) = |0\rangle_t \otimes |1\rangle_c$$
$$CZ(|1\rangle_t \otimes |1\rangle_c) = -|1\rangle_t \otimes |1\rangle_c$$

このゲームでは、＋ $|1\rangle$ と- $|1\rangle$ の違いを区別しないので、ゲームのCZ操作では、Xゲートにつながっている円の色は変わらないことがわかる。

出典：http://ibm.biz/helloq19

2.4.2. CZゲートの数理的背景



ターゲットビットを$|+\rangle$と$|-\rangle$にすると、以下のようになることが計算からわかる。

- $CZ(|+\rangle_t \otimes |0\rangle_c) = |+\rangle_t \otimes |0\rangle_c$
- $CZ(|-\rangle_t \otimes |0\rangle_c) = |-\rangle_t \otimes |0\rangle_c$
- $CZ(|+\rangle_t \otimes |1\rangle_c) = |-\rangle_t \otimes |1\rangle_c$
- $CZ(|-\rangle_t \otimes |1\rangle_c) = |+\rangle_t \otimes |1\rangle_c$

この計算と同じように、ゲームのCZは、

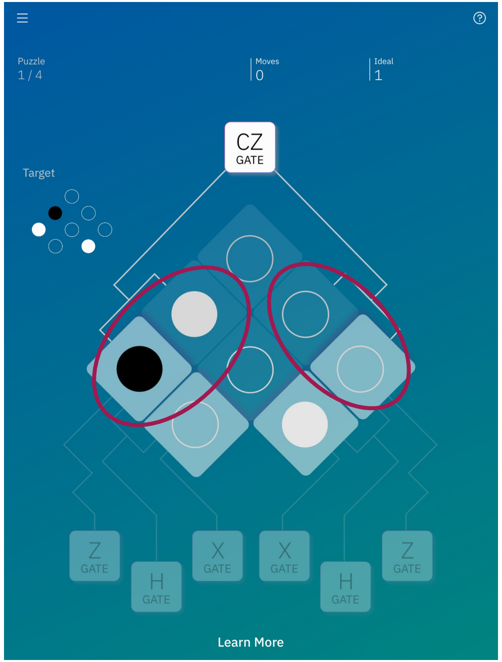
Xゲートにつながった円の片方が白の時(コントロールビットが|1>のとき)、

もう片方のZゲートにつながった円の色を反転させる

（ターゲットビットの|+>と|->を交換する）動きをすることがわかります。

出典：http://ibm.biz/helloq19

2.4.3 CZゲートの性質



どちらのビットがコントロールビットになっても、どちらのビットがターゲットビットになってもよい

コントロールビットが重ね合わせで表現されていた場合にCZゲートを作用させるとどうなるか考える。

$$CZ(|0\rangle_t \otimes |+\rangle_c) = \frac{CZ}{\sqrt{2}}\big(|0\rangle_t \otimes (|0\rangle_c + |1\rangle_c)\big) = \frac{CZ}{\sqrt{2}}(|0\rangle_t \otimes |0\rangle_c + |0\rangle_t \otimes |1\rangle_c)$$
$$= \frac{1}{\sqrt{2}}(|0\rangle_t \otimes |0\rangle_c + |0\rangle_t \otimes |1\rangle_c) = |0\rangle_t \otimes |+\rangle_c$$

$$CZ(|1\rangle_t \otimes |+\rangle_c) = \frac{CZ}{\sqrt{2}}\big(|1\rangle_t \otimes (|0\rangle_c + |1\rangle_c)\big) = \frac{CZ}{\sqrt{2}}(|1\rangle_t \otimes |0\rangle_c + |1\rangle_t \otimes |1\rangle_c)$$
$$= \frac{1}{\sqrt{2}}(|1\rangle_t \otimes |0\rangle_c - |1\rangle_t \otimes |1\rangle_c) = |1\rangle_t \otimes |-\rangle_c$$

$$CZ(|0\rangle_t \otimes |-\rangle_c) = \frac{CZ}{\sqrt{2}}\big(|0\rangle_t \otimes (|0\rangle_c - |1\rangle_c)\big) = \frac{CZ}{\sqrt{2}}(|0\rangle_t \otimes |0\rangle_c - |0\rangle_t \otimes |1\rangle_c)$$
$$= \frac{1}{\sqrt{2}}(|0\rangle_t \otimes |0\rangle_c - |0\rangle_t \otimes |1\rangle_c) = |0\rangle_t \otimes |-\rangle_c$$

$$CZ(|1\rangle_t \otimes |-\rangle_c) = \frac{CZ}{\sqrt{2}}\big(|1\rangle_t \otimes (|0\rangle_c - |1\rangle_c)\big) = \frac{CZ}{\sqrt{2}}(|1\rangle_t \otimes |0\rangle_c - |1\rangle_t \otimes |1\rangle_c)$$
$$= \frac{1}{\sqrt{2}}(|1\rangle_t \otimes |0\rangle_c + |1\rangle_t \otimes |1\rangle_c) = |1\rangle_t \otimes |+\rangle_c$$

コントロールビットとターゲットビットを入れ替えてみると、
ターゲットビットが$|1\rangle$のときに、コントロールビットの$|+\rangle$と$|-\rangle$が入れ替わる。

出典：http://ibm.biz/helloq19

3.量子プログラミングを作ってみよう

- initialize:初期状態の設定
- success_condition：最後の状態
- allowed_gates：ゲートの種類
- vi：どの量子ビットがあるか、パズルで使う円の種類、円の相関
- qubit_names：選択する量子ビットの名前
- mode：オプションで、Y出力の行を含めるかどうか

3.2 量子プログラミングの例

```
initialize = [['x', '0'],['cx', '1']]
success_condition = {'IZ': 1.0}
allowed_gates = {'0': {'h':0}, '1': {'h':0}, 'both': {'cz': 1}}
vi = [[], True, True]
qubit_names = {'0':'qubit 0', '1':'qubit 1'}
mode = None
puzzle = hello_quantum.run_game(initialize, success_condition, allowed_gates, vi, qubit_names, mode=mode)
```

- 初期状態：図1の通り
- 最終状態では、図2の通り

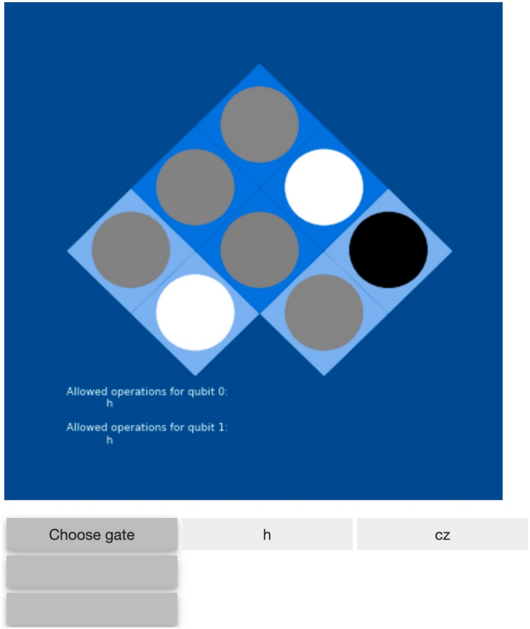


図1

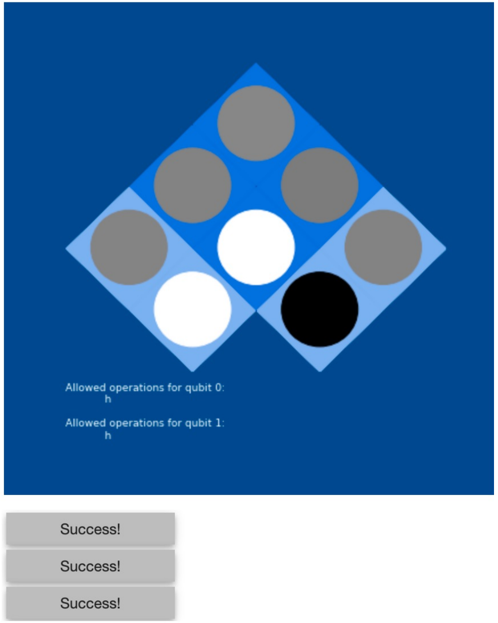
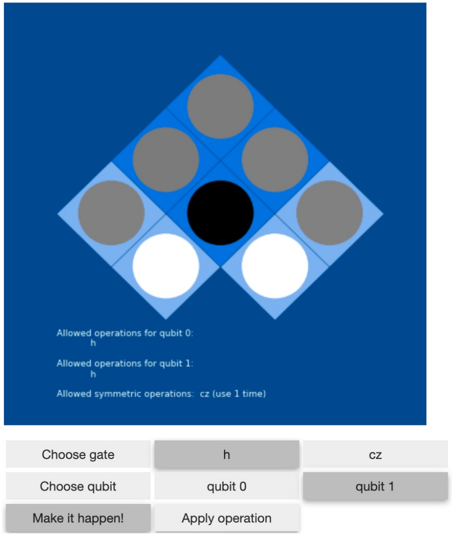
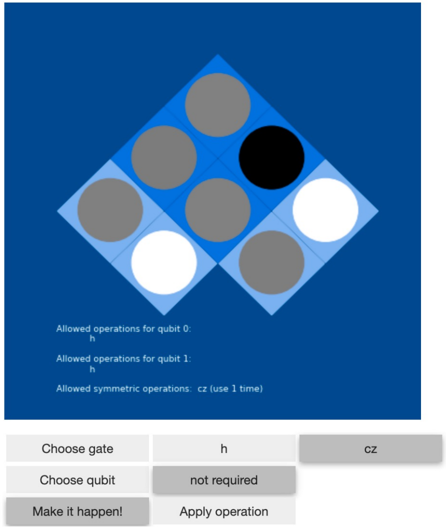


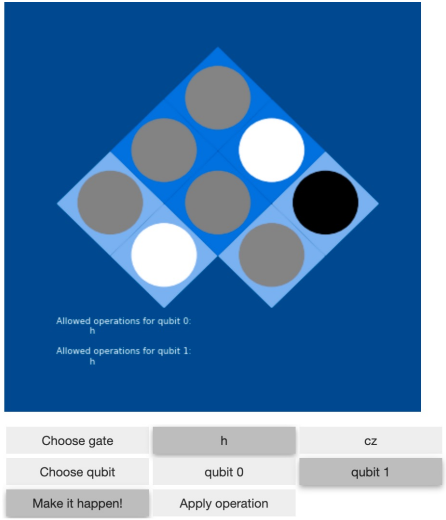
図2

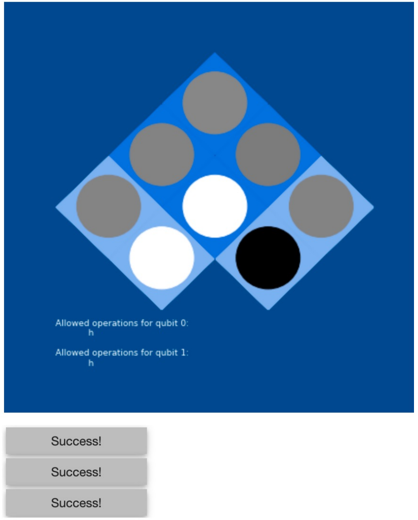
出典：https://qiskit.org/textbook/ch-ex/hello-qiskit.html#Puzzle-1

3.3 量子プログラミングの回答









Hゲート:q(1)→CZゲート→Hゲート:q(1)

4.古典ビットと量子ビットの特徴

4.1 古典ビットの特徴

1. 0または１のどちらか１つの値を取る
2. それぞれのビット値は他のビットの値とは関係なく定まる。
3. **ビットをコピーできる**

出典：https://qiita.com/testnin2/items/5e0f3a823a96c3ab0bee

4.2 量子ビットの特徴

1. **量子複製不可能 (No-Cloning) 定理**

量子ビットはビットをコピーすることができない

2. **Bell (CHSH) 不等式**

物理系が局所実在性を満たすと仮定した時に、

複数の観測系の相関の強さの上限を与える関係式

出典：https://dojo.qulacs.org/ja/latest/notebooks/1.c_CHSH-inequality_etc.html

6. 参考文献、引用文献

- Qiskit textbook
- IBMホームページ: https://www.ibm.com/blogs/research/2018/07/hello-quantum/
- 技術書典7　「量子コンピューターを理解するスマホゲーム Hello Quantum」
- Qiita『社会人のための量子コンピューター超入門 ゲート方式編 ~「Hello Quantum」はゲート方式の入り口となるか？~』

  https://qiita.com/Ayumu_walker/items/2308c76cd095faf3d04a
- 量子コンピュータと量子通信 オーム社
- 絵で見てわかる　量子コンピュータの仕組み　翔泳社
- 量子コンピューティング・量子アルゴリズムから量子機械学習まで　オーム社
- 大阪大学藤井研究室・量子コンピューティング講義資料
- Qiskit 量子プログラミング入門　Gaia教育シリーズ17
- quantum navigate dojoホームページ
- 量子コンピューター入門: 古典ビット v.s. 量子ビット:https://qiita.com/testnin2/items/5e0f3a823a96c3ab0bee

Emi ADACHI