Quantum Game Design

Esplorazione di labirinti con l’algoritmo di Grover

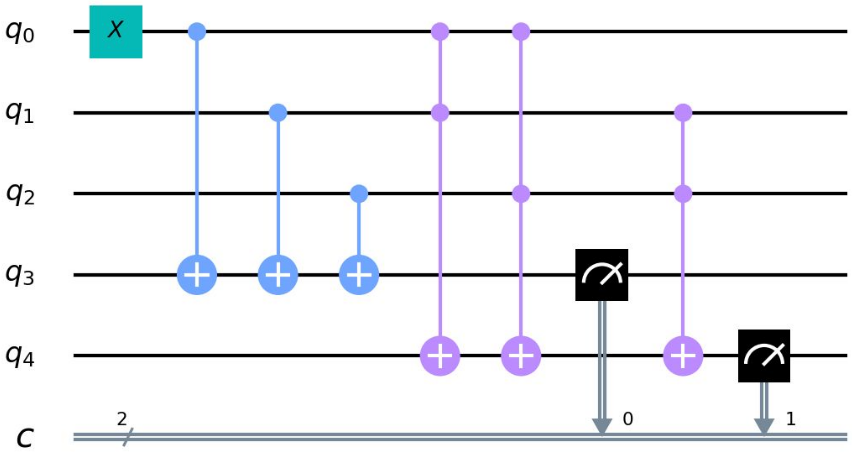
Alessio La Greca

Relatori:
Prof. Anna Bernasconi
Prof. Gianna M. Del Corso
Dott. Alessandro Berti

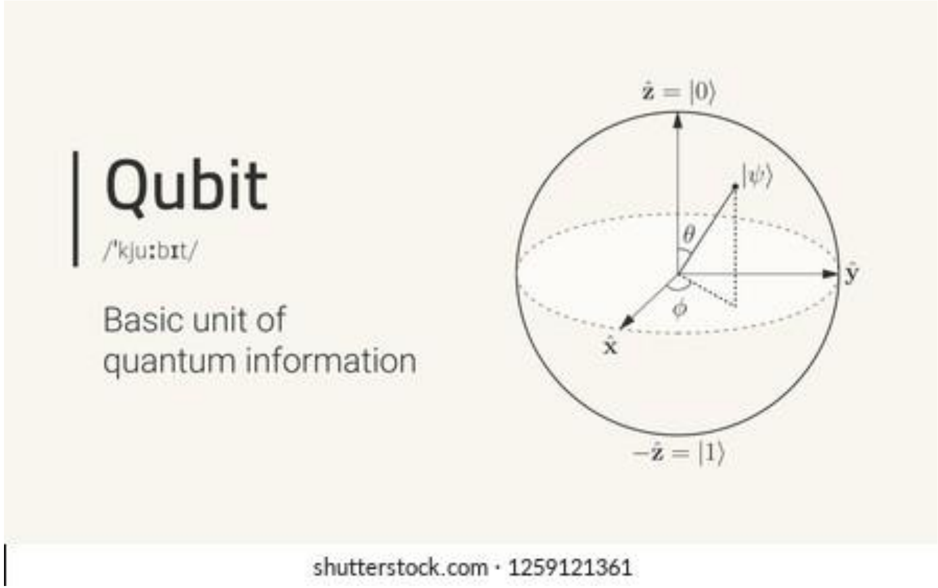
Controrelatore:
Prof. Roberto Bruni

Videogiochi e Quantum Computing:

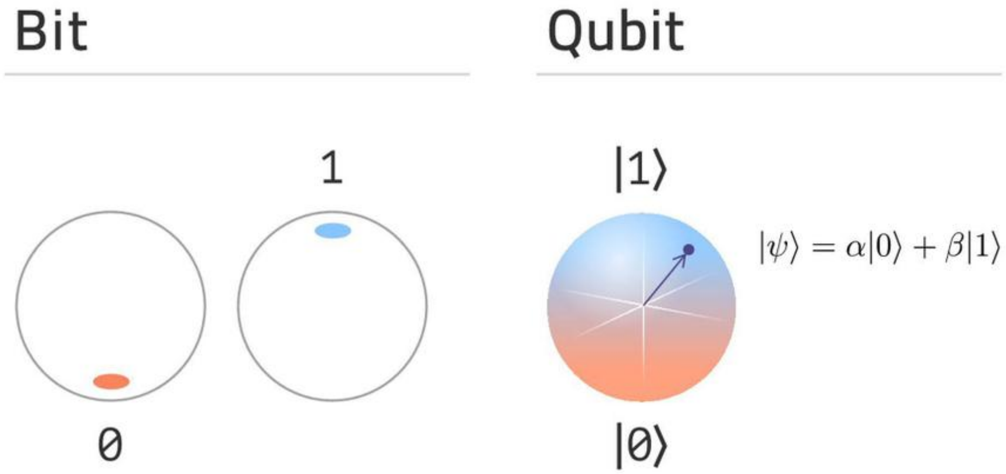
Perfetti sconosciuti o promessi sposi?



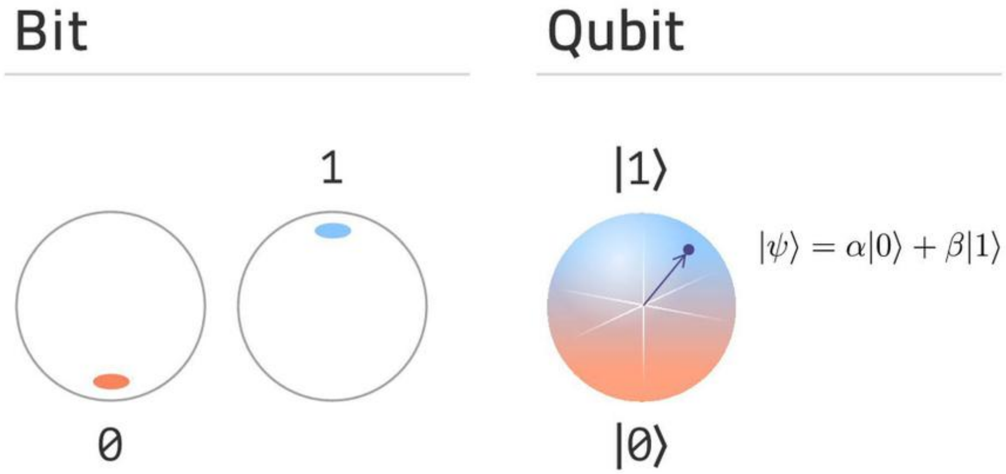
1. Qubit e porte



Nel Quantum Computing l’unità di informazione fondamentale è il “quantum bit”, o qubit.



Nel Quantum Computing l’unità di informazione fondamentale è il “quantum bit”, o qubit.



un qubit = $|q\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle\,,\ \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1.$

Possiamo **misurare** un qubit per ottenere un bit. Abbiamo:

- $|\alpha|^2$ probabilità di ottenere un bit nello stato 0
- $|\beta|^2$ probabilità di ottenere un bit nello stato 1

Possiamo manipolare i qubit usando delle porte...

Se i qubit sono vettori, le porte sono matrici!

Solitamente un qubit parte dallo stato $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Porta $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ $\qquad X|q\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$

Porta $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ $\qquad H|q\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{\alpha+\beta}{\sqrt{2}} \\ \frac{\alpha-\beta}{\sqrt{2}} \end{bmatrix}$

Porta $\text{C-NOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ $\qquad \text{C-NOT}\,|q_1 q_2\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{bmatrix}$

Può essere generalizzata con più qubit di controllo nella *porta Toffoli*

In generale, un sistema a *n* qubit viene rappresentato da un vettore di dimensione 2^n

2. L’algoritmo di Grover


TIENIMI LA MANO
UNA PASSEGGIATA NELLA NOTTE
CACCIATORI DEL MARE
P.D. JAMES
©Unamicainviaggio

?


ANIMAL
you can do it


TIENIMI LA MANO
CACCIATORI DEL MARE
UNA PASSEGGIATA NELLA NOTTE
P.D. JAMES
©Unamicainviaggio

?

VS

?

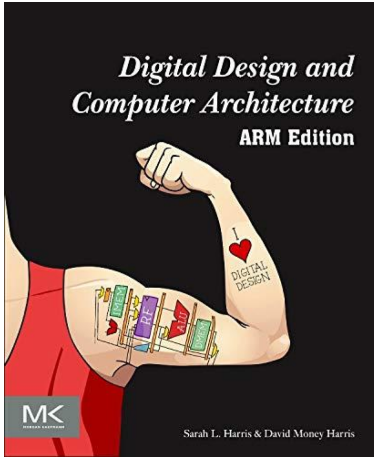
IN SUPREMÆ DIGNITATIS
1343


©Unamicainviaggio

?

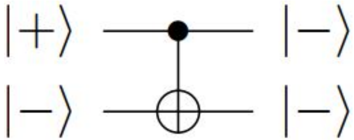
VS


Google
Digital Design and Computer Architecture ARM Edition


Digital Design and
Computer Architecture
ARM Edition
I
DIGITAL DESIGN
MK
Sarah L. Harris & David Money Harris

La porta C-NOT ha un qubit di *controllo* e uno *target*...

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \; |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$



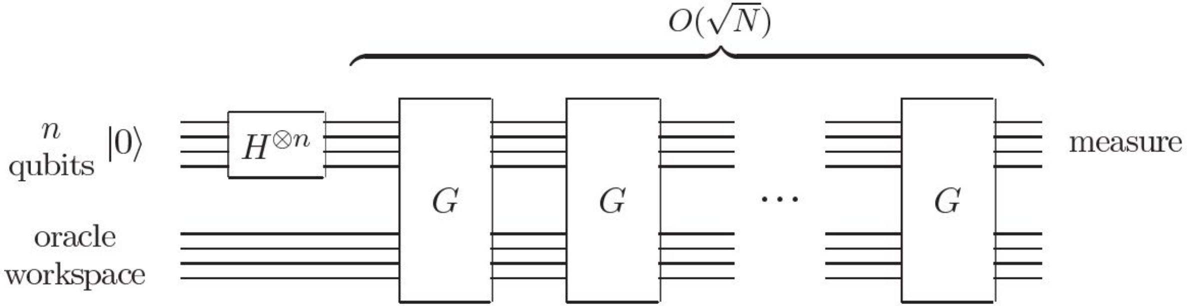
...ma il target *può* modificare il qubit di controllo!

Questo fenomeno è conosciuto col nome di *kickback di fase*

A Grover piace
questo elemento...

Spazio di ricerca composto da *N* elementi $\Rightarrow$ dobbiamo usare un sistema a $log_2(N) = n$ qubit

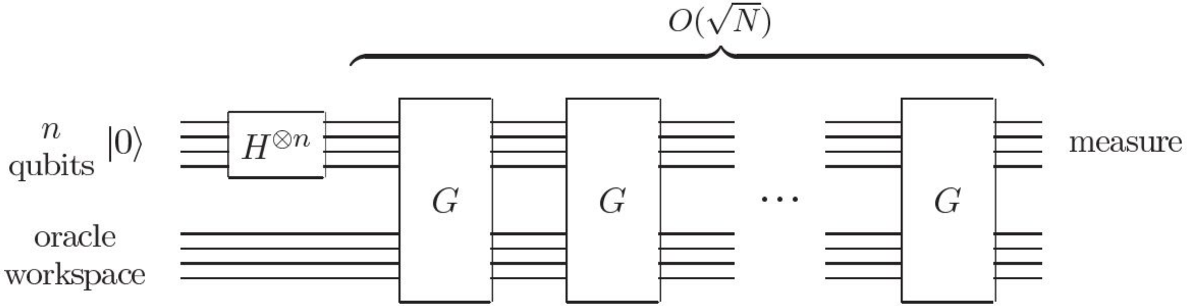
Algoritmo =



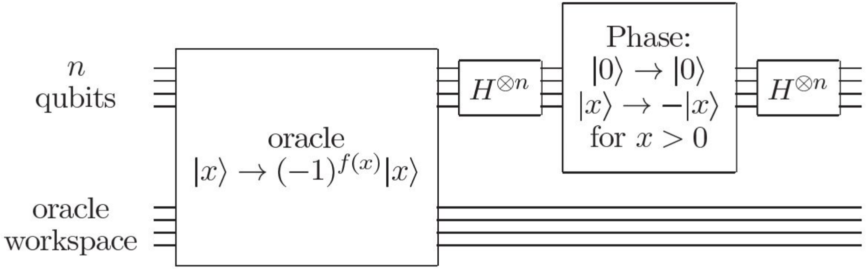
1) Si pongono gli *n* qubit in uno *stato di ugual sovrapposizione* usando porte H.

Spazio di ricerca composto da N elementi $\Rightarrow$ dobbiamo usare un sistema a $log_2(N) = n$ qubit

Algoritmo =



G =



1) Si pongono gli n qubit in uno *stato di ugual sovrapposizione* usando porte H.

2) Per $O(\sqrt{N})$ volte si applica G, ovvero:

   2.1) Si applica l'oracolo O che riconosce la soluzione quando la vede.

   2.2) Si applica il kickback di fase -1 alla soluzione (diffuser).

È dimostrabile algebricamente che con i giusti calcoli dobbiamo applicare l'operatore G $O(\sqrt{N})$ volte...

$$|\psi\rangle = \frac{1}{\sqrt{4}}\sum_{x=0}^{3}|x\rangle = \frac{1}{\sqrt{4}}|00\rangle + \frac{1}{\sqrt{4}}|01\rangle + \frac{1}{\sqrt{4}}|10\rangle + \frac{1}{\sqrt{4}}|11\rangle = \frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

$$\sum_x{}' |x\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \qquad \sum_x{}'' |x\rangle = \begin{bmatrix}1\\1\\1\\0\end{bmatrix},$$

$$2|\alpha\rangle\langle\alpha| - I = 2\frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}\frac{1}{\sqrt{3}}\begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix} - I = \frac{2}{3}\begin{bmatrix}1 & 1 & 1 & 0\\1 & 1 & 1 & 0\\1 & 1 & 1 & 0\\0 & 0 & 0 & 0\end{bmatrix} - \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}-\frac{1}{3} & \frac{2}{3} & \frac{2}{3} & 0\\\frac{2}{3} & -\frac{1}{3} & \frac{2}{3} & 0\\\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} & 0\\0 & 0 & 0 & -1\end{bmatrix}$$

$$|\beta\rangle = \frac{1}{\sqrt{1}}\sum_x{}' |x\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \qquad |\alpha\rangle = \frac{1}{\sqrt{3}}\sum_x{}'' |x\rangle = \frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}$$

$$|\phi'\rangle = (2|\alpha\rangle\langle\alpha| - I)|\phi\rangle = \begin{bmatrix}-\frac{1}{3} & \frac{2}{3} & \frac{2}{3} & 0\\\frac{2}{3} & -\frac{1}{3} & \frac{2}{3} & 0\\\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} & 0\\0 & 0 & 0 & -1\end{bmatrix}\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\-1\end{bmatrix}$$

$$2|\psi\rangle\langle\psi| - I = 2\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix} - I = \frac{2}{4}\begin{bmatrix}1 & 1 & 1 & 1\\1 & 1 & 1 & 1\\1 & 1 & 1 & 1\\1 & 1 & 1 & 1\end{bmatrix} - \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}-\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2}\end{bmatrix}$$

$$\text{Proiezione di } |\psi\rangle \text{ su } |\alpha\rangle = \left(\frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}\frac{1}{\sqrt{3}}\begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}\right)\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1 & 1 & 1 & 0\\1 & 1 & 1 & 0\\1 & 1 & 1 & 0\\0 & 0 & 0 & 0\end{bmatrix}\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}$$

$$\cos(\frac{\theta}{2}) = \|\text{Proiezione di } |\psi\rangle \text{ su } |\alpha\rangle\| = \left\|\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\0\end{bmatrix}\right\| = \sqrt{\frac{1}{4}+\frac{1}{4}+\frac{1}{4}} = \sqrt{\frac{3}{4}} = \sqrt{\frac{4-1}{4}}.$$

$$\text{Proiezione di } |\psi\rangle \text{ su } |\beta\rangle = \left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}\right)\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \frac{1}{\sqrt{4}}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

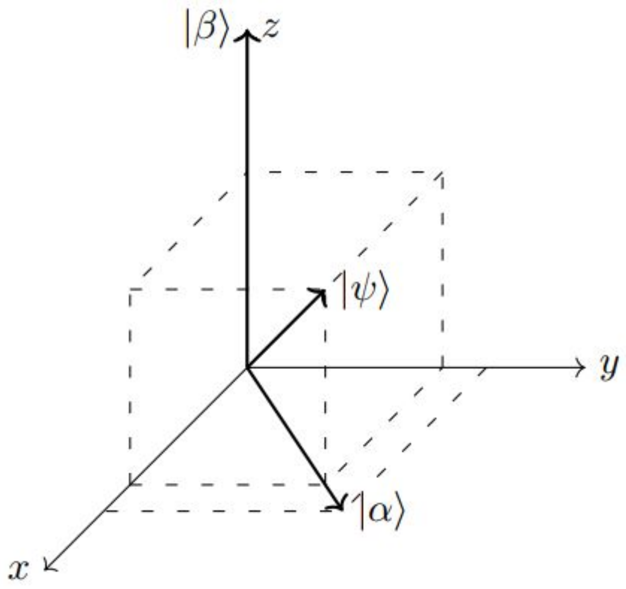
$$\sin(\frac{\theta}{2}) = \|\text{Proiezione di } |\psi\rangle \text{ su } |\beta\rangle\| = \left\|\frac{1}{\sqrt{4}}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right\| = \sqrt{\frac{1}{4}}.$$

$$|\phi''\rangle = (2|\alpha\rangle\langle\alpha| - I)|\phi'\rangle = \begin{bmatrix}-\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2}\\\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2}\end{bmatrix}\frac{1}{\sqrt{4}}\begin{bmatrix}1\\1\\1\\-1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

$|\alpha\rangle$ = *vettore di ugual sovrapposizione dei fallimenti.*

$|\beta\rangle$ = *vettore di ugual sovrapposizione dei successi.*

$|\psi\rangle$ = *vettore di ugual sovrapposizione degli stati.*

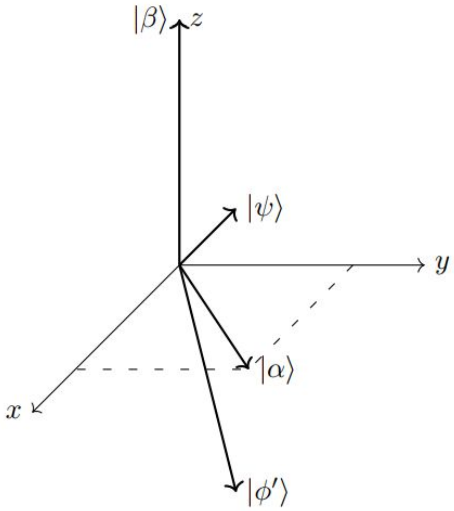


Quando misuriamo il sistema di qubit otteniamo o un fallimento del vettore $|\alpha\rangle$ o un successo del vettore $|\beta\rangle$. Misureremo con maggiore probabilità una possibile soluzione del vettore più vicino a $|\psi\rangle$.

Possiamo portare $|\psi\rangle$ vicino a $|\beta\rangle$? Sì, ruotandolo come si deve!

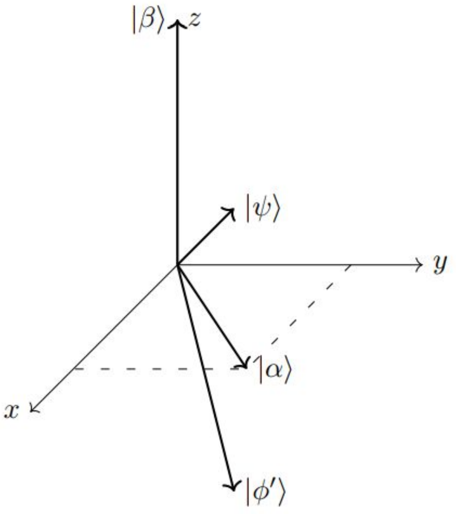
(Ribattezziamo $|\psi\rangle$ come $|\phi\rangle$)

L'oracolo di Grover riflette $|\phi\rangle$ rispetto a $|\alpha\rangle$, ottenendo $|\phi'\rangle$:

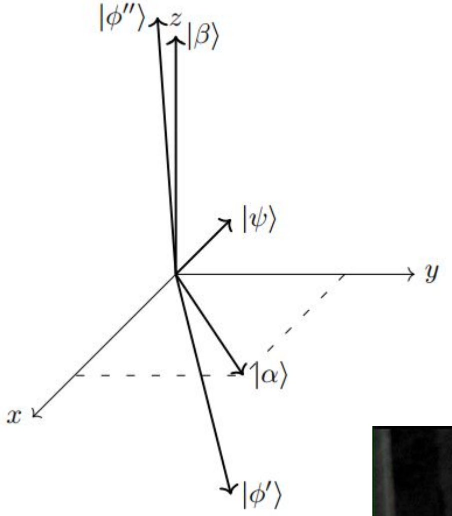


(Ribattezziamo $|\psi\rangle$ come $|\phi\rangle$)

L'oracolo di Grover riflette $|\phi\rangle$ rispetto a $|\alpha\rangle$, ottenendo $|\phi'\rangle$:



Il diffuser riflette $|\phi'\rangle$ rispetto a $|\psi\rangle$, ottenendo $|\phi''\rangle$:

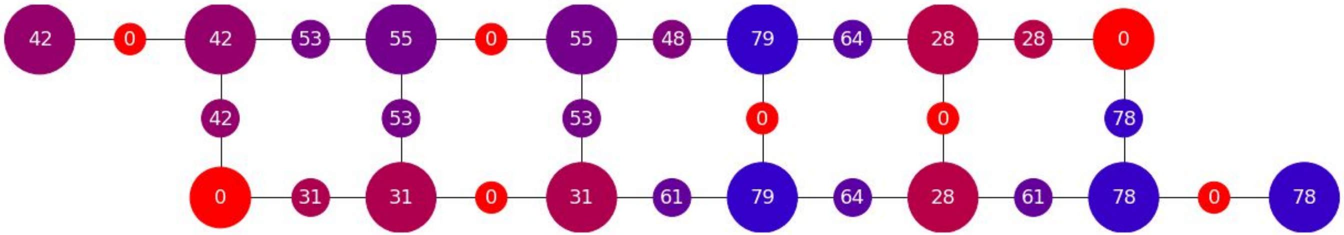


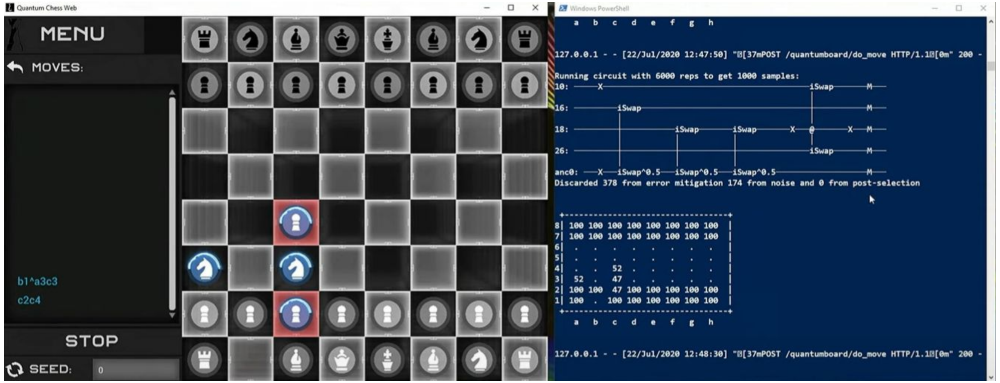
Doppia riflessione = rotazione.

Le rotazioni sono matrici, ma le matrici sono porte!

3. Videogiochi e Quantum

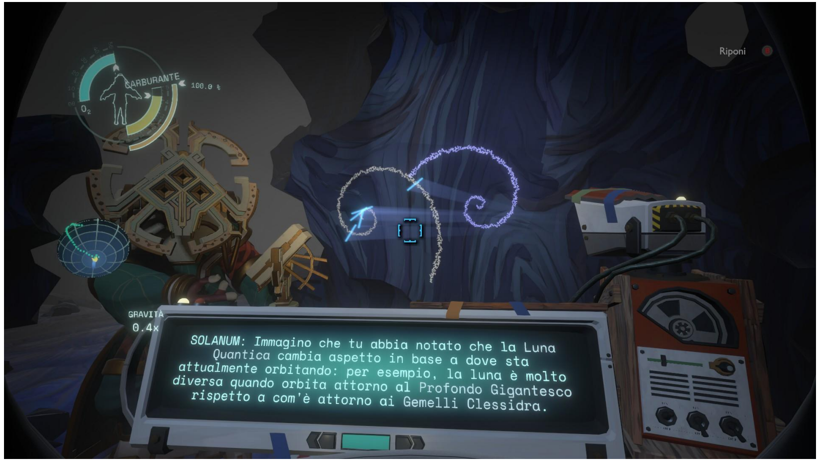
Quantum Awesomeness: testiamo le performance di un computer quantistico con un gioco.





Quantum Chess: sfruttiamo i principi della meccanica quantistica per costruire un gioco.

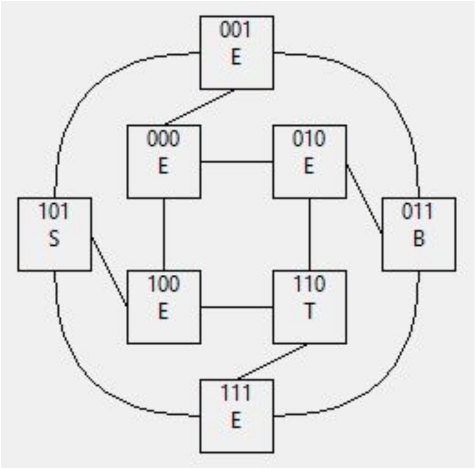
Outer Wilds: facciamo conoscere la fisica quantistica a chiunque con un gioco d'avventura.



4. Esplorazione di labirinti con Grover

Abbiamo fatto alcune assunzioni semplificative per limitare la complessità (comunque 25+ qubit):

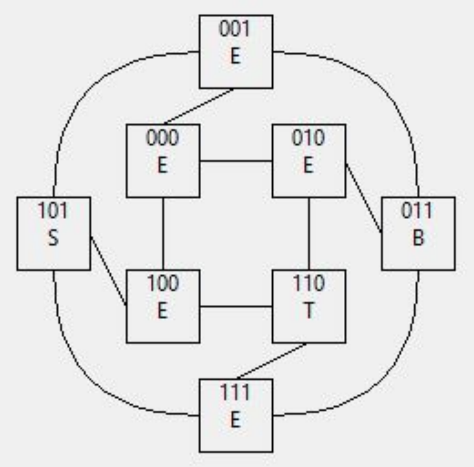
- -8 stanze identificate da un indice a tre bit.
- -si parte sempre dalla stanza di indice 000.
- -ogni stanza è connessa ad altre tre.
- -nemici, tesoro, negozio o boss.
- -per ogni potenziamento, ne esiste un'istanza a partita.



4. Esplorazione di labirinti con Grover

Abbiamo fatto alcune assunzioni semplificative per limitare la complessità (comunque 25+ qubit):

- -8 stanze identificate da un indice a tre bit.
- -si parte sempre dalla stanza di indice 000.
- -ogni stanza è connessa ad altre tre.
- -nemici, tesoro, negozio o boss.
- -per ogni potenziamento, ne esiste un'istanza a partita.



Ad esplorarlo è un agente:

- -con 4 punti salute.
- -con un punto attacco.

Tesoro e negozio aumentano le statistiche $\Rightarrow$ maggiori chance di battere il boss!

$$\text{probabilità di vittoria} = 5 \times (1 + \text{salute}) \times \text{attacco}.$$

	attacco = 1	attacco = 2	attacco = 3	attacco = 4
salute = 1	10%	20%	30%	40%
salute = 2	15%	30%	45%	60%
salute = 3	20%	40%	60%	80%
salute = 4	25%	50%	75%	100%

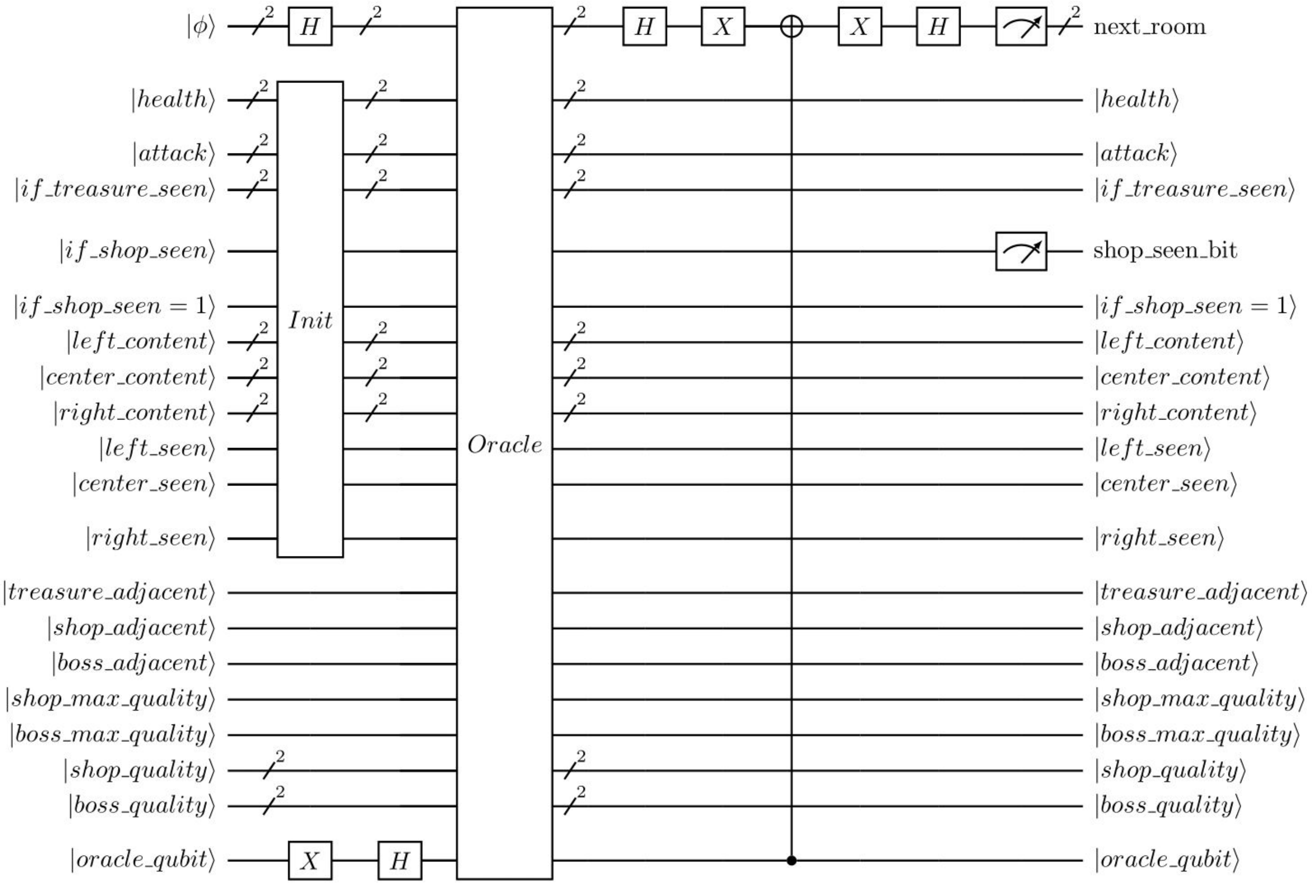
-sappiamo solo quello che abbiamo intorno.
-vogliamo trovare un percorso che termina col boss e massimizza le probabilità di vittoria.
-ad ogni partita il contenuto di ciascuna stanza cambia.

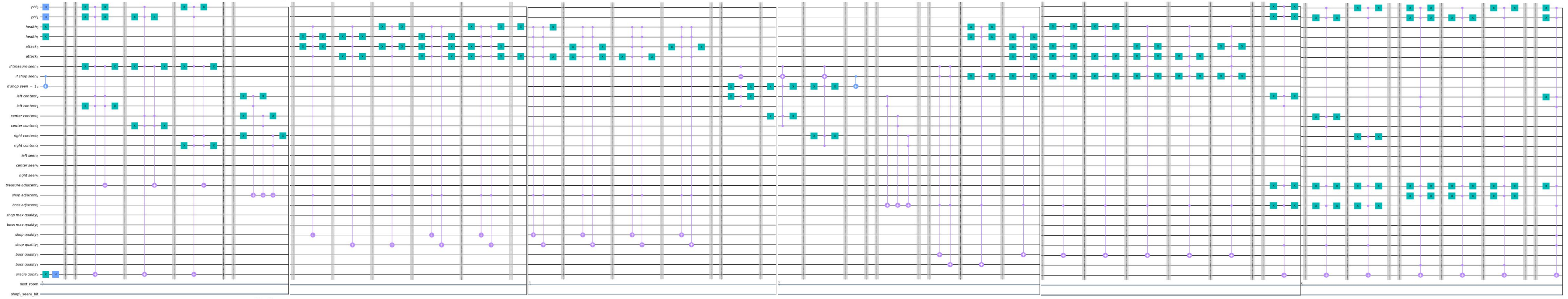

Sempre cercare il tesoro

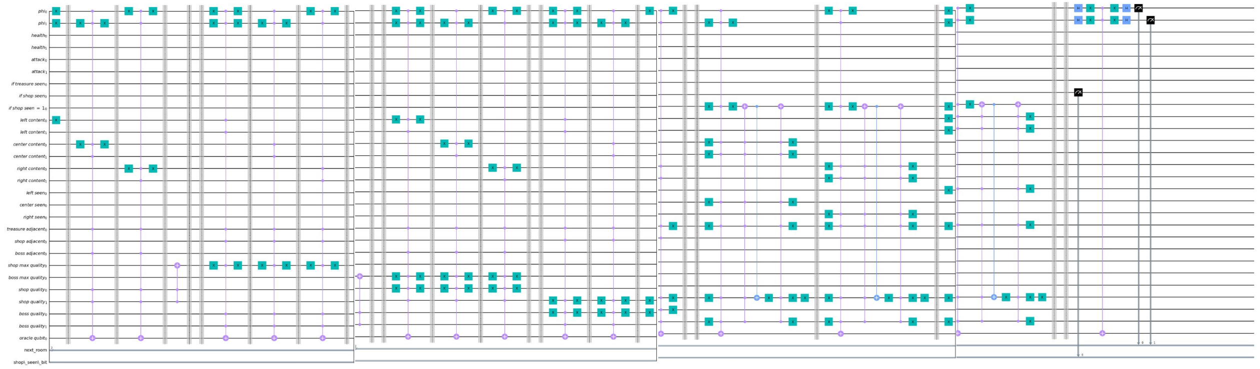

Evitare se possibile i nemici

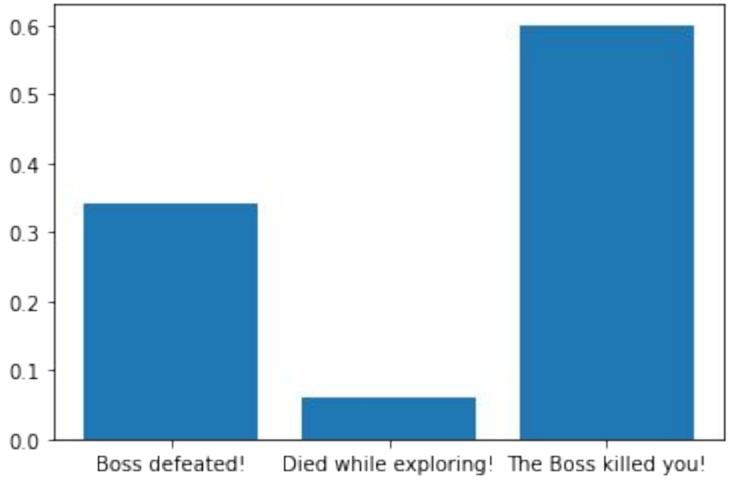

Valutare se vale la pena
esplorare il negozio

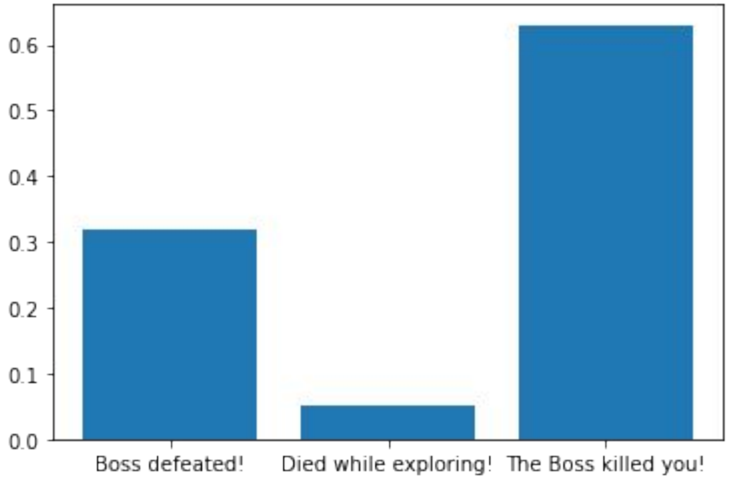
Studiando la struttura del problema abbiamo costruito l’oracolo (e quindi il circuito) giusto...




phi0
phi1
health0
health1
attack0
attack1
if treasure seen0
if shop seen0
if shop seen = 10
left content0
left content1
center content0
center content1
right content0
right content1
left seen0
center seen0
right seen0
treasure adjacent0
shop adjacent0
boss adjacent0
shop max quality0
boss max quality0
shop quality0
shop quality1
boss quality0
boss quality1
oracle qubit0
next_room
shop_seen_bit


phi0
phi1
health0
health1
attack0
attack1
if treasure seen0
if shop seen0
if shop seen = 10
left content0
left content1
center content0
center content1
right content0
right content1
left seen0
center seen0
right seen0
treasure adjacent0
shop adjacent0
boss adjacent0
shop max quality0
boss max quality0
shop quality0
shop quality1
boss quality0
boss quality1
oracle qubit0
next_room
shop_seen_bit





Algoritmo classico

Complessità: O(N)

Algoritmo quantistico

Complessità: $O(\sqrt{N})$

Grazie per l’attenzione!