QUANTUM TECHNOLOGY
Hello Quantum Developers World

Yet Another Frontier for JavaScript
Hello Quantum Developers World

QCon London 2020
Miguel Ramalho
Quantum Spectrum

Physicists
Theorise

Scientists
Empirically test

Engineers
Implement

Software Engineers
Connect

Developers
Work Hard...
Goals

Physicists

Demystify

Scientists

Engineers

Software Engineers

Hands dirty

Developers

Benefit

bit.ly/qcon-qjs
Quantum Computing is...

leveraging *natural phenomena*

happening at the *atomic* and *subatomic* scale

to produce computations
2 energy states (simplification)
these are quantized: take only discrete (quantified) values.

Hydrogen atom
1 electron orbiting the nucleus
2 energy states (simplification)
Any two valid quantum states can be combined (superposed) into another valid state.
Circuit Model

| $|x_0\rangle$ | $A$ | $|y_2\rangle$ |
|-------------|-----|---------------|
| $|x_1\rangle$ |     | $|y_1\rangle$ |
| $|x_2\rangle$ | $B$ | $|y_0\rangle$ |

$H$, $R_4$, $X$
Circuit Model (Components)

Qubit
Classical bit
Gate
Measurement

Gate
NOT (Pauli-X)
Hadamard

Notation

Time
Tools
Quantum JavaScript libraries

In this presentation

- Quirk
- Qiskit
- Quantum Peep

For reference

- Jsqubits
- Qiskit-js
- JsQuil
- IBM QX
Quirk

Drag and drop web app to test Quantum Circuits
Circuit Model in Quirk

Bloch Sphere
Qiskit
Framework for Quantum Computing
Maintained by IBM
Quantum Peep

JS Tool to create and execute Quantum Circuits

github.com/mapmeld/quantum-peep

```
> npm i quantum-peep
```

```
const qp = require('quantum-peep');
const viz = require('quantum-circuit-viz');
```
Hello World Circuit

```javascript
let p = new qp.Program();
p.add(qp.Gates.X(0))
p.measure(0, 0);
p.measure(1, 1);
console.log(viz.textViz(p));
```

BasicGate { qubits: [0], inverse: false, name: 'X' }

Measure { qubit: 0, register: 0, name: 'measure' }

Measure { qubit: 1, register: 1, name: 'measure' }

```
q_0: |0>  
q_1: |0>  
c_0: 0  
c_1: 0
```
Quantum Peep

Works with

github.com/mapmeld/quantum-peep
Quantum Peep

```
let q = new qp.RigettiProcessor({
  api_key: 'no_need',
  user_id: 'qcon'
});

q.run(p, runTimes, (res)=>console.log(res));
q.devices((device) => {
  // [{ "name": "ibmq_ourense", "status": "on", ...}]
});
```

```
let q = new qp.IBMProcessor({
  login: "username",
  token: "api_token",
  processor: 'ibmqx4'
});
```
Q Measurement

```javascript
let p = new qp.Program();
p.add(qp.Gates.X(1));
p.add(qp.Gates.X(1));
p.measure(1, 1);
p.add(qp.Gates.X(0));
p.measure(0, 0);
console.log(viz.textViz(p));
```
Q Superposition Measurement

```java
p = new qp.Program();
p.measure(1, 1);
p.add(qp.Gates.H(0));
p.measure(0, 0);
q.run(p, 1000, (result) => {
  console.log(result);
});
```
Q Entanglement

```javascript
p = new qp.Program();
p.add(qp.Gates.H(0));
p.add(qp.Gates.CX(0, 1));
p.measure(1, 1);
p.measure(0, 0);
console.log(viz.textViz(p));
```
True Random Numbers

How would you do this?
Classical

Quantum
Parallelism
Given $N$

Find primes $p, q$ such that: $N = p \times q$
Shor’s Algorithm
15 = 3 x 5
1099551473989 = 1048589 x 1048601
Quantum Approximate Optimization Algorithm (QAOA)

Find approximate solutions for constraint satisfaction problems

- Traveling salesman problem
- Scheduling
- Graph Max-cut
Solve optimization problems

Quantum Tunnelling

Adiabatic evolution
Quantum Supremacy
Quantum Supremacy

A moment in time, when quantum computers can perform more complex computations than classical computers can simulate.
Quantum

Cloud
Quantum Cloud Services

Join the quantum revolution
Let’s talk about how quantum can help you solve problems and energize your innovation.
Rigetti Computing

A quantum-classical cloud platform optimized for variational hybrid algorithms
QC Ware

user interface

Problem/Data Mapping

high-level features

Anneal Offset
Decomposition

algorithms

Binary Optimization
Chemistry Simulation
Machine Learning

hardware

Annealing
Circuit Model
Simulators

D-Wave
IBM
Rigetti
IBM
Google
Rigetti
Microsoft
Qiskit Aqua (Algorithms for QUantum computing Applications)
Takeaways for Quantum Developers

**Today**, you can

- use JS to learn (visually and code)
- invoke real quantum processors for free
- Use AI and optimization techniques on small problems
Takeaways for Quantum Developers

**Tomorrow**, you should (might want to)

- Beware of traditional encryption
- Understand how to map hard problems into Q-solvable ones

For now, we must wait for Quantum Supremacy
Summary

Physicists
Demystified

Scientists

Engineers

Software Engineers
Dirty Hands

Developers
Benefit
Thank you

Question Time

bit.ly/qcon-qjs
Thank you

Question Time

by Miguel Ramalho

@msramalho

bit.ly/qcon-qjs
Additional Resources

- Google’s Quantum-Computing-Playground
- Google’s Cirq
- Qutech @ TU Delft’s Quantum Inspire
- QC Ware’s Forge