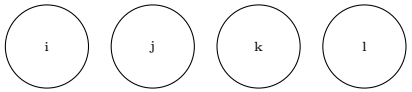
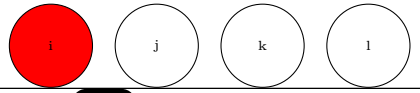
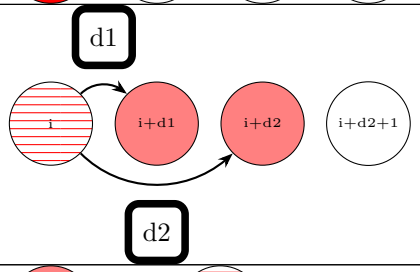
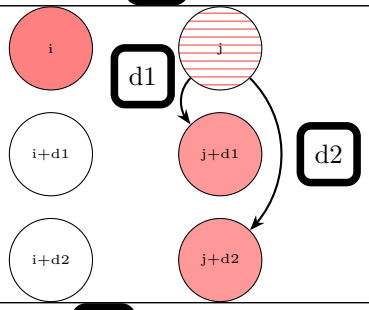
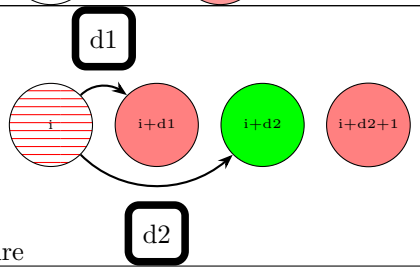
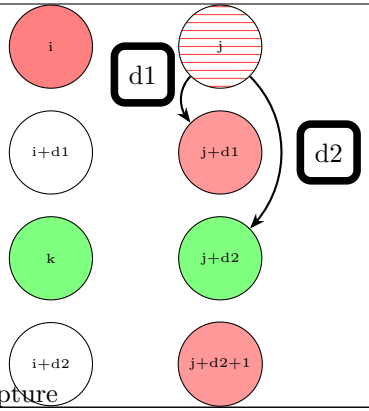
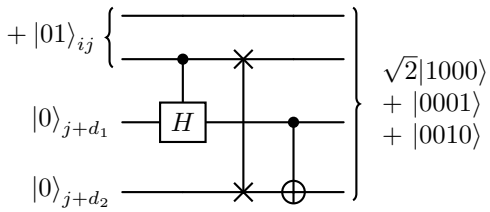
Move name	Move	Python list	Quantum array	Qiskit
Initialize		board=[None] * 32	$\|0\rangle_i$ — $\|0\rangle_j$ — $\|0\rangle_k$ — $\|0\rangle_l$ —	q = QuantumRegister(10, 'q') circuit = QuantumCircuit(q)
New pawn		board[i] = { 'color' : red, 'probability' : 1, 'pawn' : 1 }	$\|0\rangle_i$ — X — $\|1\rangle_i$ $\|0\rangle_j$ —— $\|0\rangle_k$ —— $\|0\rangle_l$ ——	circuit.x[q[i]]
First move		board[i] = None board[i+d1, i+d2] = { 'color' : red, 'probability' : 0.5, 'pawn' : 1}	$\|1\rangle_i$ — ● — ✕ — $\|0\rangle$ $\|0\rangle_{i+d_1}$ — H — ✕ — } $\|01\rangle + \|10\rangle$ $\|0\rangle_{i+d_2}$ — ⊕ —	circuit.ch(q[i], q[i+d1]) circuit.swap(q[i], q[i+d2]) circuit.cx(q[i+d1], q[i+d2])
Second move		board[i, j] = None board[$i + d_{1,2}, j + d_{1,2}$] = {'color' : red, 'probability' : 0.25, 'pawn' : 1}	$\|10\rangle_{ij} + \|01\rangle_{ij}$ { ● — ✕ — $\|0\rangle_{j+d_1}$ — H — ✕ — } $\sqrt{2}\|1000\rangle$ + $\|0001\rangle$ + $\|0010\rangle$ $\|0\rangle_{j+d_2}$ — ⊕ —	circuit.ch(q[i], q[i+d1]) circuit.ch(q[j], q[j+d1]) circuit.swap(q[i], q[i+d2]) circuit.swap(q[j], q[j+d2]) circuit.cx(q[i+d1], q[i+d2]) circuit.cx(q[j+d1], q[j+d2])
full-full capture		board[i] = None board[i+d1, i+d2+1] = { 'color' : red, 'probability' : (1/2), 'pawn' : 1}	$\|1\rangle_i$ — ● — ✕ — $\|0\rangle$ $\|0\rangle_{i+d_1}$ — H — ✕ — } $\|110\rangle + \|001\rangle$ $\|1\rangle_{i+d_2}$ — ⊕ — ● $\|0\rangle_{i+d_2+1}$ — ✕ — ⊕ —	circuit.ch(q[i], q[i+d1]) circuit.swap(q[i], q[i+d2]) circuit.cx(q[i+d1], q[i+d2]) circuit.cx(q[i+d2+1], q[i+d2])

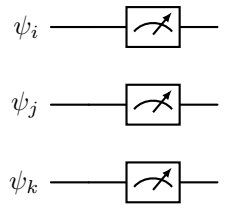


1 half-half capture

board[i, j] = None
board[$i + d_{12}, j + d_1$,
$j + d_1 + 1$] = {
'color' : red,
'probability' : 1/4,
'pawn' : 1}




```
circuit.ch(q[i], q[i+d1])
circuit.swap(q[i], q[i+d2])
circuit.cx(q[i+d1], q[i+d2])
circuit.cx(q[i+d2+1], q[i+d2])
```




```
for i in range(0,c.len()):
  if c[i] == 0:
    board[i] = None
  else:
    board[i]['probability'] = 1
```

Measure

```
for (q_i, c_i) in zip(q,c):
  circuit.measure(q_i, c_i)
```