Exposing the Quantum Circuit with a Children's Card Game: Quantum 6 Nimmt!

Victor Chang
ippouzutsu@ucla.edu

March 2021

Abstract

This paper details the game of Quantum 6 Nimmt!. Quantum 6 Nimmt! takes the concept of the quantum circuits and integrates it with the simple game of *6 Nimmt!*.[1] This game's purpose is to familiarize players with the quantum circuit and primitive quantum operations.

1 Introduction

With quantum computing on the rise, it might be prudent to have an understanding quantum phenomena. Games are a good way to learn new concepts. This paper demonstrates how to take the card game of *6 Nimmt!* and modify its rules to transform it Quantum 6 Nimmt!. The paper will also detail some findings on the performance of Qiskit's state vector simulation of a quantum system. An offline proof of concept of the game has been implemented in Python with Qiskit's state vector simulation used to implement the quantum operations of the game.

1.1 Other Related Works

Chris Cantwell's Quantum Chess serves as the main inspiration of the game [1]. It takes the quantum concepts of superposition, entanglement, and interference and integrates those concepts with chess. In quantum chess, pieces on the board occupy a space on the board with a probability, and moves in that game manipulate said probabilities. Quantum 6 Nimmt! has a similar formulation where something in the game is considered a probability, and these probabilities can be manipulated by the players to their advantage in a quantum manner.

2 Base Game Outline

Here, the rules of *6 Nimmt!* will be outlined mathematically. If one wishes to see examples of gameplay, there are plenty of gameplay videos on YouTube. The unconventional notation $o.a$ shall be used and defined as such: $o.a$ is the attribute, a, of an object, o, the same notation found in object oriented code. Also for future reference, the term "board" and "rows" will be used interchangeably to mean the same thing.

2.1 *6 Nimmt!* Objects

The game of *6 Nimmt!* can have 2-10 players, P. Each player, p, starts with 66 points denoted $p.p$. There are 104 cards, denoted as the set, C, numbered from 1 to 104. Each card, $c \in C$, in addition to its number, $c.n$, has a point value associated to the card which is deducted from players based on who collected the cards. The point value, $c.v$, is determined by the number of the card and ranges from 1 to 7.[2] The board consists of 4 rows, R. A row, $r \in R$, holds a list of cards, $r.C$, that have been discarded each turn. Each row holds at least 1 card and holds up to 5 cards. So framing the game objects mathematically:

[1] wikipage here: https://en.wikipedia.org/wiki/6_Nimmt!

[2] Details on how point values are determined are in *6 Nimmt!*ś wikipage. They remain unchanged in Quantum 6 Nimmt!.

- The set of players $P : \{p \mid \ p.p \leq 66\}$, $2 \leq |P| \leq 10$.
- The set of cards $C : \{c \mid 1 \leq c.n \leq 104,\ 1 \leq c.v \leq 7,\ c.v, c.n \in \mathbb{Z}\}$, $|C| = 104$. All $c.n$ are distinct.
- The set of rows (i.e. the board) $R : \{r \mid 1 \leq |r.C| \leq 5, \forall c \in r.C, c \in C\}$

2.2 *6 Nimmt!* Rules

The game is played in rounds. Each round, 10 cards are randomly dealt to each player to form each player's hand, $p.C$. Each row starts with one card drawn from the remaining deck. Each turn all players must discard exactly 1 card. When all players have made their selection, the discards, D, are ordered by their number from least to greatest and appended one by one whilst keeping track of each card's owner, $c.p$. A card appends the row whose last card's number is the maximum of all rows whose last card's number is less than the card. If no such card exists, the player must select a row to collect, with their discard becoming the first and only card of the new row. The row is also replaced by the discarded card if the card happens to be 6th card to be appended.

A new round is started when all cards in hand have been discarded and no one has reached 0 points. Players try to retain as many points as possible before someone reaches 0 points. On the round that a player reaches 0 points, the round is played through until all cards are discarded from the players' hands. The players who retain the most points are declared the winners, W. The game can be formulated as an algorithm where the input is group of players playing the game and the return value is the set of winners. The pseudo-code of the algorithm is as follows (two backslashes denote comments within the pseudo-code):

Algorithm 6Nimmt(P) - the game of 6Nimmt with a set of players P

```
for all p ∈ P do
   p.p := 66
end for
while ∀p ∈ P, p.p > 0 do
   initialize the deck C \\ (as described earlier)
   deal 10 cards from C to every player p ∈ P
   initialize R and append each row r.C with one card from the remaining cards in C
   while ∀p ∈ P, p.C ≠ ∅ do
      D := {each player p's selected discard c, c ∈ p.C}
      sort(D, key = c.n) \\ c ∈ D
      for all c ∈ D do
         candidate_rows := {r ∈ R | (r.C's last card).n < c.n}
         if candidate_rows = ∅ then
            discard's owner c.p selects a row r to take
            replace_row(c, r)
         else
            candidate_row := arg max_r({c.n | c ∈ candidate_rows' last cards, r ∈ candidate_rows})
            if |candidate_row.C| = 5 then
               replace_row(c, candidate_row)
            else
               append candidate_row.C with c
            end if
         end if
      end for
   end while
end while
max_score := max({p.p | p ∈ P})
W := {p ∈ P | p.p = max_score}
return W
```

Algorithm replace_row(c, r) - Applies the penalty on player $c.p$ for taking a row of cards r replaces the cards in the row $r.C$ with the single element list containing c. Here we assume $c.p$ points to the player p.

```
for all c ∈ r.C do
    c.p.p −= c.v
end for
r.C := [] \\ empty the list
append c to r.C
```

2.3 Modification Before Making *6 Nimmt!* Quantum

To make *6 Nimmt!* quantum, we must consider something as a probability. When a row is replaced, it is certain that the player who caused the replacement loses points based on how many points are in the row. For now we add a biased coin, q, to each row (so $r.q$) with two possible outcomes. The first outcome (we will call this "0") is the same as in the base game, points are deducted from the player who replaced the row. The second outcome (we will call this "1") is defined such that all players aside from the player who replaced the row will have their points deducted instead. The sum of the deductions will be equal to however many points would have been deducted from one person in the first outcome. This is so that on average, the change on the total of all players points with each row replacement is the same as before. In the base game, every row's coin is biased to have the probability of the outcome "0" to be 1. After the calculations, we keep row replacement the same. We modify the replace_row() procedure as such:

Algorithm replace_row(c, r) - Deducts points according to the outcome of $r.q$.

```
outcome := flip r.q
if outcome = 0 then
    for all c ∈ r.C do
        c.p.p −= c.v
    end for
else
    P′ := P − {c.p}
    for all p ∈ P′ do
        for all c ∈ r.C do
            p.p −= c.v/|P′|
        end for
    end for
end if
r.C := [] \\ empty the list
append c to r.C
```

3 Quantum 6 Nimmt! Outline

After a simple modification, we can transform this game into something that involves quantum. The quantum circuit will be introduced which will be used to determine how scores are calculated and a few more modifications will be made to the game to make Quantum 6 Nimmt!.

3.1 Brief Primer on Quantum

In quantum computing we have the qubit. The qubit has the two basis states of $|0\rangle$ and $|1\rangle$ which can be considered 0 and 1 respectively for the sake of simplicity. To place a qubit in superposition is like flipping a coin, and not observing the outcome until one desires to do so. Upon observing the outcome (measurement), the outcome persists until another quantum operation is applied onto the qubit. Quantum operations can not only affect the state of individual qubits, but it can also enforce correlations between qubits on systems with multiple qubits via entanglement.

The question could be asked: Could we just keep the coins instead of relying on qubits and quantum operations? Theoretically, one could come up with a rule set that manipulates these probabilities on the coins without having to think in terms of quantum. Thinking in terms of quantum operations, however, gives us a predefined way to manipulate these probabilities without having to define new rule set for manipulating probabilities. Quantum operations have a nice property where successive quantum operations (more precisely, unitary operations) without measurement can be reversed by applying the inverse of all the quantum operations since they preserve information of the previous state. This is unlike classical computing where a lot of operations on bits are irreversible. For example, the AND of 0 and 1 is 0, but there is no way to tell whether both bits were 0 or not without looking at the original input. In Quantum 6 Nimmt!, all available quantum operations are reversible in some way, so players can potentially cancel each other's actions. The collapsing of the quantum system due to measurement is actually beneficial as it gives an inherent way to reduce the probability state space which can be utilized by a player to their advantage. This will be elaborated on promptly in the next section where we introduce the quantum components to the game.

3.2 Introducing the Quantum Components to the Game

Now let us introduce the quantum components to the game of *6 Nimmt!* to make it Quantum 6 Nimmt!.

Each row r has a qubit $r.q$ associated with it. The measurement of this qubit will be the "coin" to determine how the scores are calculated when the row has been replaced by a player. All $r.q$ will make up the quantum circuit Q where each $r.q$ can be considered a wire on Q. A new quantum circuit is initialized to the state of $|0000\rangle$ at the start of each round. Throughout the course of the round the state of the quantum circuit will change such that in general:

$$Q = \alpha_{|0000\rangle} |0000\rangle + \alpha_{|0001\rangle} |0001\rangle + ... + \alpha_{|1111\rangle} |1111\rangle$$

$$\sum_{\alpha} |\alpha|^2 = 1$$

The generalization describes the quantum system as a probability distribution on all possible states of the quantum system if one were to measure all the qubits at the same time. However, in the game only one qubit is ever measured at a time, so for a player, it is easier to consider qubits of interest and what operations have affected said qubits. Every time a measurement occurs the, probability distribution collapses based on the qubit observed. For example, say we have this 2-qubit quantum system:

$$Q = \alpha_{|11\rangle} |11\rangle + \alpha_{|00\rangle} |00\rangle$$

We measure the first qubit and its outcome is $|1\rangle$. The resulting quantum system becomes:

$$Q = |11\rangle$$

The collapsing of the quantum system can signify to all players that the other row with the second qubit will be $|1\rangle$ even though it has not been measured. Understanding the collapse of quantum states is one of the keys to help a player win.

Quantum operations affect the qubit that they are applied to. They are chosen by the player to be associated to the card the player discard's during that turn. They shall be denoted $c.o$ (o for operation) and upon being appended to a row r, $c.o$ shall be applied to $r.q$ if it is a single qubit operation. $c.o$ will also affect the qubit of the row chosen by the player if $c.o$ is a two-qubit operation. The available set of quantum operations include: H, Z, $CNot$, and $hSwap$.

- H is the Hadamard gate. This is the rudimentary operation used to place qubits into a superposition. If it is applied to a qubit whose current state is either $|0\rangle$ or $|1\rangle$, the qubit is placed into a superposition where measuring the qubit is becomes an unbiased coin flip between $|0\rangle$ and $|1\rangle$.
- Z is the Pauli-Z gate. It does not affect $|0\rangle$ and maps $|1\rangle$ to $-|1\rangle$. The motivation for its inclusion is when H, Z, H is applied to a qubit in succession, it toggles the qubit (i.e. $HZH|0\rangle = |1\rangle$ and $HZH|1\rangle = |0\rangle$ since $HZH = X$, X being the Pauli-X gate which toggles the qubit).

- $CNot$ is the controlled NOT gate. It is an operation that affects 2 qubits where one is considered the "control" and the other is considered the "target". It toggles the target qubit when the control is $|1\rangle$. This is the rudimentary operation for entanglement.
- $hSwap$ is $\sqrt{Swap}$. An intuitive explanation of what it does is it performs the standard $Swap$ operation on two qubits half-way (hence $hSwap$ for "half of a $Swap$"). Thus, two successive $hSwap$ operations will result in a regular $Swap$ operation. By itself, it can be thought of as an equalizer on two qubits. It is termed $hSwap$ for the sake of brevity.

Note that H, Z, and $CNot$ are their own inverses, meaning if the same operation is applied to the same qubit in succession, the operations cancel each other out. 4 successive $hSwap$s on the same qubits (with source and target remainining the same throughout) will cancel each other out. The player can also elect not to have any quantum operation associated with the qubit. This would be equivalent to placing the identity gate I, which does not do anything to the quantum circuit.

3.3 Quantum 6 Nimmt! Gameplay Rules

Given the rules of base *6 Nimmt!* we add a few more rules to finish the definition of the Quantum 6 Nimmt! rules.

1. Each player at the start of each round will start with 2 "Mqops", denoted as $p.mq$. "Mqops" stands for multi-qubit operations. Each player starts with 2 every round. This resource denotes how many more multi-qubit operations ($CNot$ and $hSwap$) a player can apply to a card during the round.
2. Any time a row r is replaced, its respective qubit $r.q$ is measured. So replace "flip $r.q$" with "measure $r.q$" in line 1 of the modified replace_row() operation. $|0\rangle$ and $|1\rangle$ are considered 0 and 1 respectively.
3. For single qubit operations, when a card is going to be appended onto a row r, the quantum operation $c.o$ is applied to $r.q$.
4. For 2-qubit operations, when a card c is going to be appended onto a row's list of cards $r.C$, the player is prompted to select another row. If either of the rows are full (i.e. have 5 cards), both of the rows are replaced (therefore, both of their qubits are measured) before appending the operation. Otherwise, the shorter row is appended with filler cards such that both rows have contain the same number of cards. Any filler cards that are needed have $c.v = 0$. This operation will be noted as column_sync(). The last card appended to the target row inherits the number of what was previously the last card before the column_sync operation. $r.C$'s last card will still be c. Finally, the operation is applied to the two qubits, $r.q$ serves as the source, while the target row's qubit serves as the target.
5. When a card's number is too low, (i.e. $candidate\_rows = \emptyset$), we ignore the quantum operation on the card by overwriting the operation ($c.o$) with the empty string, such that when the card is appended, no operation is applied on the quantum circuit.

The pseudo-code for Quantum 6 Nimmt! is described on the following pages. It can be noted that if all players never applied a quantum operation to any card during the whole game, then the result would simply just be base *6 Nimmt!*.

Algorithm Quantum6Nimmt(P) - the game of Quantum 6 Nimmt! with a set of players P

for all $p \in P$ **do**
    $p.p := 66$
end for
while $\forall p \in P,\ p.p > 0$ **do**
    $p.mq := 2$
    initialize the 4-qubit quantum circuit Q with each $r.q$ pointing to one qubit in Q
    initialize the deck C
    deal 10 cards from C to every player $p \in P$
    initialize R and append each row $r.C$ with one card from the remaining cards in C
    while $\forall p \in P,\ p.C \neq \emptyset$ **do**
        $D :=$ {each player p's selected discard $c,\ c \in p.C$}
        sort(D, $key = c.n$) \\ $c \in D$
        for all $c \in D$ **do**
            if $c.o \in \{CNot,\ hSwap\}$ **then**
                $c.p.mq \mathrel{-}= 1$
            end if
            $candidate\_rows := \{r \in R \mid (r.C\text{'s last card}).n < c.n\}$
            if $candidate\_rows = \emptyset$ **then**
                $c.o :=$ '' \\ overwrite to be no operation
                discard's owner $c.p$ selects a row r to take
                replace_row(c, r)
            else
                $candidate\_row := \arg\max_r(\{c.n \mid c \in candidate\_rows\text{' last cards}, r \in candidate\_rows\})$
                if $c.o \in \{CNot,\ hSwap\}$ **then**
                    $target\_row :=$ another row selected by player $c.p$
                    $target\_lc :=$ card t with $t.v = 0$, $t.n = target\_row's$ last card's number, $t.p = c.p$
                    if $|candidate\_row.C| = 5$ or $|target\_row| = 5$ **then**
                        replace_row(c, $candidate\_row$)
                        replace_row($target\_lc$, $target\_row$)
                    else
                        column_sync($candidate\_row.C$, $target\_row.C$)
                        append $candidate\_row.C$ with c
                        append $target\_row.C$ with $target\_lc$
                    end if
                else
                    if $|candidate\_row.C| = 5$ **then**
                        replace_row(c, $candidate\_row$)
                    else
                        append $candidate\_row.C$ with c
                    end if
                    apply $c.o$ on $candidate\_row.q$
                end if
            end if
        end for
    end while
end while
$max\_score := \max(\{p.p \mid p \in P\})$
$W := \{p \in P \mid p.p = max\_score\}$
return W

Algorithm replace_row(c, r) - Deducts points according to the outcome of $r.q$. (Quantum 6 Nimmt!)

$outcome$:= measure $r.q$
if $outcome = 0$ **then**
  for all $c \in r.C$ **do**
    $c.p.p \mathrel{-}= c.v$
  end for
else
  $P' := P - \{c.p\}$
  for all $p \in P'$ **do**
    for all $c \in r.C$ **do**
      $p.p \mathrel{-}= c.v/|P'|$
    end for
  end for
end if
$r.C$:= [] \\ empty the list
append c to $r.C$

4 Evaluation

In terms of scalability, a 4-qubit system is trivial to simulate. The state vector representation of the system would be a 16-element vector where each element is a complex number. Every quantum operation on the system is the same as multiplying a 16x16 matrix to the vector to return a vector of the same dimensions. As a result, application of the quantum operations and measurement take milliseconds for the game. Though it can be mentioned that no interface for successive measurements and operations was found.

```
Application of quantum operations finished in 0.009997367858886719 seconds.
Measurement finished in 0.008984804153442383 seconds.
Application of quantum operations finished in 0.0009810924530029297 seconds.
Measurement finished in 0.0 seconds.
Application of quantum operations finished in 0.0 seconds.
Measurement finished in 0.0 seconds.
Application of quantum operations finished in 0.005999326705932617 seconds.
Measurement finished in 0.0010199546813964844 seconds.
Application of quantum operations finished in 0.0009832382202148438 seconds.
Measurement finished in 0.0009999275207519531 seconds.
```

Figure 1: Typical execution times of applying and measuring the simulated quantum circuit of a Quantum 6 Nimmt! game.

Whether a beginner can learn about quantum while playing this game remains to be seen. *6 Nimmt!* itself is a game that is better played than explained. A more beginner friendly version of the game would include the probabilities of each qubit measuring $|1\rangle$ so that the effects of each quantum operation are made more apparent. The proof of concept does not do so, as it assumes that it is the player's responsibility to keep track of said probabilities.

5 Motivations for Various Choices

5.1 Why 6 Nimmt!?

6 Nimmt! is a fair game where everyone is dealt a random set of cards. One card is chosen randomly by each player each turn, and the order in which operations are applied is determined by the ascending order of the discards's number each turn as in base 6 Nimmt. Thus, every player is subject to the same non-determinism. The way the board state is structured carries some similarities to how a quantum circuit is constructed. As a result, 6 Nimmt! was chosen to be augmented with a rule set that includes quantum computation.

5.2 Determining the Set of Quantum Operations

I chose to exclude X for fear of the game devolving into players simply using X instead of using H. H, Z, and $CNot$ are rudimentary operations in quantum computing that are simple to understand. These operations are their own inverses, so when two of the same operation affect the same qubit(s) in succession, the two operations cancel out. $hSwap$ highlights more weirdness of quantum without being too nebulous to comprehend. Theoretically, one could change the available operations to be a universal gate set, then any unitary quantum operation would be possible.

5.3 "Column Syncing" during the Application of Multi-Qubit Operations

Column syncing is to ensure the chronology of operations on the quantum system is better reflected in the board state. It is more of an aesthetic choice to include this procedure, and it can be removed if undesired.

6 Conclusion

Hopefully, Quantum 6 Nimmt! can be included in the minuscule library of quantum games. Maintaining a 4-qubit quantum system is definitely feasible, especially with the existing quantum computers having 10s of qubits. In the end, Quantum 6 Nimmt! is simply *6 Nimmt!* with extra steps. A beginner could simply start out throwing random quantum operations whilst the expert could wreak havoc on every other player's point total by exploiting the quantum system to his/her advantage.

References

[1] Christopher Cantwell. Quantum chess: Developing a mathematical framework and design methodology for creating quantum games, 2019.