Application of game theory to software user interface evaluation

Chidozie C. Nwobi-Okoye1* and Stanley Okiy2

Abstract: The choice of a software userinterface is very strategic. Usability testing alone often does not determine the best software user interface. Secondly, economic considerations and the fact that software performance is a stochastic dynamic process (Markov process), are often neglected in user interface evaluations. Hence, evaluation tools that better capture the overall user experience, in addition to user interface economics and dynamics, should evaluate software user interfaces more effectively. The psychological concept of flow has been proposed as better metric for measuring software user experience. Hence, a good software interface design should have flow characteristics, a virtual strategy. The strategic nature of software user interface necessitates the need for the best tool for modeling the strategic interaction among competing software firms/user interfaces. The decision problem therefore, is how to develop an appropriate game theoretic model to assist software userinterface designers. In the light of these shortcomings, game theoretic model using markov chains and the flow theory concept was used to model and obtain optimal/equilibrium software user interface among competing firms/user-interfaces. The results of the study show that game theoretic analysis would be a very useful tool for software user interface evaluation.

ABOUT THE AUTHORS
Dr Chidozie C. Nwobi-Okoye is lecturer in the Faculty of Engineering of Chukwuemeka Odumegwu Ojukwu University, Nigeria. His research interest include: Operations Research, CAD/CAM, Game Theory, Simulation and Artificial Intelligence.

Dr Stanley Okiy is a training officer in the department of welding engineering and offshore technology, Petroleum Training Institute, Effurun, Nigeria. He is involved in active academic research and engineering consultancy. His research interest include: Operations research and project management.

PUBLIC INTEREST STATEMENT
Usually rival software companies compete to design software which users like better than the other. They try to add features that would improve engagement and user experience. Such features often known as flow features are usually added to make the software use flow.

When flow features are added to software, users are not bored and time seems to fly when in use, giving users a false sense of reality known as virtual reality. Software firms often employ this strategy to gain market share over rival firms, giving rise to strategic interaction among them.

In order to have a better understanding of the scientific principles behind such interaction, the theory of virtual games was used to model the strategic interaction. The results gave an insight to what decisions and actions software firms need to take in product design in order to retain their market share, continue to be competitive and remain in business.
1. Introduction
Making a software usable is costly, the problem is therefore how to balance cost and benefit of increased usability and the general user experience. The three pillars to the success of a software application are function, features, and face. Function answers the question: Does the software perform as intended? Features answer the question: Does the software meet user’s requirement? Face answers the question: Is the software usable? It is obvious that the choice of a user interface is very strategic, hence the decision problem is how to develop an appropriate game theoretic model to assist software user interface designers.

Evaluating alternative software user interfaces is a daunting task. Usability testing is one way of evaluating alternative user interfaces (Tractinsky, Katz, & Ikar, 2000). Tractinsky et al. (2000) found out that aesthetics contributes significantly to the overall user experience. Hence, a software that is less usable but beautiful could be ranked on the same level of user experience with a more usable but less beautiful one. Even the software user interface's response to human emotions play very important role in user experience. Good designers usually consider the emotional impact of their design, and this concept has been defining the design of some software user interfaces (Picard & Jonathan, 2002). Thus usability testing alone cannot ascertain better software user interfaces. The major concepts of usability are:

Efficiency: Once users have learned the design, how quickly can they perform tasks?
Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

While one could measure efficiency, learnability, memorability and errors, it is very difficult to measure satisfaction, which is intrinsically linked to software features and function. This is where the concept of flow comes in to user interface studies in human computer interaction. The flow concept better captures the overall user experience. In the very competitive modern world better user interfaces constitute unique selling points for products, the success of Apple’s Iphone and Google’s Android are typical examples. These two mobile operating systems software together hold 99.7 percent of the smartphone market in the first quarter of 2017 (International Data Cooperation (IDC), 2017). According to Ramon Llamas, senior research analyst at IDC, some of the features in Iphone and Android had existed in other smartphone platforms, but the way Apple and Google made smartphone experience intuitive and seamless has quickly earned a massive following. According to Nayebi et al. (2012), the fact that complex computing systems are finding their way into everyday life en mass makes usability very critical making firms to develop their products with user oriented methods instead of technology oriented methods. It means that the best user interface design constitute an equilibrium strategy for competing firms. Hence, game theory becomes the natural tool for determining the equilibrium strategy.

It follows from above that the success of a software application depends significantly on the user interface design. Hence, an optimum user interface design must always be sought. A good software user interface design should be able to induce waiting, elicit high concentration from users, make users have subjective sense of time etc. These are flow characteristics. These attributes are necessary because many times the user has to wait while the software queues for system resources or retrieves information or data, or sometimes some tasks become frustrating and unpleasant to users etc. Koufaris (2002) applied flow theory to study and analyse online consumer behaviour. Kay McMahon in his M.Sc. dissertation at Robert Gordon university investigated the influence of website usability on business (McMahon, 2005). These researchers found out that e-business sites that are usable have flow features and are more likely to attract and retain customers more than sites that
does not have flow features. His work was one of the attempts to introduce the concept of flow into software design. A number of other researchers have in recent years studied flow in relation to information technologies and human computer interaction. These writers recommended flow experience as a possible way of measuring online consumer experience (Ghani & Deshpande, 1994; Hoffman & Novak, 1996; Novak, Hoffman, & Yung, 1998; Trevino & Webster, 1992; Webster, Trevino, & Ryan, 1993). In general, therefore, a good software user interface must have flow features. Once a software user interface has flow characteristics, time seems to fly which is in accordance with the work of the psychologist Mihály Csíkszentmihályi (Csíkszentmihályi, 1990), who proposed the concept of flow to describe such phenomenon. Csíkszentmihályi described flow as “being completely involved in an activity for its own sake. The ego falls away. Time flies. Every action, movement, and thought follows inevitably from the previous one, like playing jazz. Your whole being is involved, and you’re using your skills to the utmost” (Geirland, 1996).

From the foregoing it is apparent that any strategy (scheme) used to incorporate flow characteristics in software user interfaces is a virtual strategy (Nwobi-Okoye, 2009, 2010a). Such schemes could be animations, aesthetics, music, emotion boosters etc. Apart from the technical aspect of software user interfaces, the economic aspect which plays a very strategic role, and the fact that software performance is a stochastic dynamic process (Markov process) are often neglected.

Game theory by its nature is a very powerful tool for evaluating alternative strategies for the purpose of selecting the optimum or most efficient strategy (Nwobi-Okoye & Igboanugo, 2011). This makes game theory amenable to performance evaluation. Hence game theory has been applied to performance evaluation in various fields of life. In the field of electronics and communication engineering, game theory is used for network performance evaluation. For instance, Shifat, Al Zobery, Rahman, and Chowdhury (2015) used a game theoretic framework based algorithm and priority based access with dynamic channel allocation scheme for ensuring better signal to interference and noise ratio (SINR) to deal with the reduction of outage probability in a heterogeneous (HetNet) network. Hamdoun, Rachedi, Tembine, and Ghamri-Doudane (2016) used evolutionary game theory for the design of efficient transmission strategy selection algorithm for M2M communications underlaying cellular networks. Simulation results of their study showed that the evolutionary game based transmission strategy selection algorithm avoids significant degradation of traditional human-to-human (H2H) services in terms of throughput and fairness compared to a single non-cooperative game strategy. For application in supply chain management, Eskafi, Roghanian, and Jafari-Eskandari (2015) used cooperative game theory and evolutionary game theory for designing a performance measurement system for supply chain. Jalali Naini, Aliahmadi, and Jafari-Eskandari (2011) used evolutionary game theory for designing a mixed performance measurement system for environmental supply chain management. Tchangani (2006) used game theory for group evaluation of production units. He evaluated the performance of each decision unit through the satisfiability functions in the framework of satisficing game theory. Sun et al. (2012a, 2012b) used applied cooperative game theory to feature evaluation and selection in from high-dimensional data-set. They used a cooperative game theory based framework to evaluate the power of each feature. The results they obtained showed that the proposed algorithm achieves better results than other methods in most cases. Jahangoshai Rezaee, Moini, and Haji-Ali Asgari (2012), Jahangoshai Rezaee, Moini, and Makui, (2012) through their research showed that game theory could be used for performance evaluation of health facilities and power plants.

From the foregoing, it is apparent that evaluation of software user interfaces could be done effectively using game theory. The purpose of this paper, therefore, is to apply the game theoretic model developed by Nwobi-Okoye (2009) to analyse strategic software user interface design.
2. Background

2.1. Flow concept and virtual reality
Flow is the mental state of operation in which the person is very fully immersed in what he or she is doing by a feeling or sense of energized focus or attention, full involvement, and success in the process of the activity. Flow could be defined as the loss of perception of time when concentrating or immersed in an activity. The concept of flow, a positive psychology concept, which was proposed by Mihály Csíkszentmihályi (Csíkszentmihályi, 1990) has been widely referenced across a variety of fields.

According to Csíkszentmihályi, flow is completely focused motivation. It is regarded as a single-minded immersion and represents perhaps the ultimate in harnessing the emotions in the service of performing and learning. In flow the emotions are not just contained and channeled, but are at the same time positive, charged, and completely aligned with the task at hand. Being caught in the acedia of depression or boredom is to be barred from flow experience. The hallmark of flow is a feeling of spontaneous joy, even rapture, while performing a task (Goleman, 1990).

Some factors identified by Csíkszentmihályi as accompanying an experience of flow are (Csíkszentmihályi, 1990):

1. Clear goals (expectations and rules are discernible and goals are attainable and align appropriately with one’s skill set and abilities). Moreover, the challenge level and skill level should both be high.
2. Concentrating, a high degree of concentration on a limited field of attention (a person engaged in the activity will have the opportunity to focus and to delve deeply into it).
3. Loss of the feeling of self-consciousness, the merging of action and awareness.
4. Distorted sense of time, one's subjective experience of time is altered.
5. Direct and immediate feedback (successes and failures in the course of the activity are apparent, so that behavior can be adjusted as needed).
6. Balance between ability level and challenge (the activity is neither too easy nor too difficult).
7. A sense of personal control over the situation or activity.
8. The activity is intrinsically rewarding, so there is an effortlessness of action.
9. People become absorbed in their activity, and focus of awareness is narrowed down to the activity itself, action awareness merging.

Virtual reality occurs when the payoff determining factors assume certain conditions exist which in fact do not. Virtual reality strategies use deceptive perceptions to improve payoffs for the strategist (Nwobi-Okoye, 2009, 2010a, 2010b, 2010c, 2011).

When there is loss of perception of time when one is concentrating, time flies. This is virtual reality because in the real sense of the world, time never flies, it is just perception. Software with flow features is thus more likely to attract more customers than the ones without flow features.

2.2. Virtual games
According to the theory of virtual games, virtual games are games where the competitors or players use strategies known as virtual strategies (Nwobi-Okoye, 2009, 2010a). Virtual reality occurs when the payoff determining factors or agents assume certain conditions exist which in fact do not. For example the use of modern virtual reality (VR) gadgets makes users experience conditions which seem to exist but in fact don’t. Virtual reality strategies use deceptive perceptions to improve payoffs for the strategist (Nwobi-Okoye, 2009, 2010a). Mathematically a virtual game is defined thus:
2.2.1. Definition
A virtual game is defined as a game with a finite set of players \( i \in J \), a grand/secondary payoff matrix, \( G_G \), a set of virtual strategies, \( V = (V_1, V_2, V_3, \ldots V_n) \) for each player with each strategy tied to a virtual payoff matrix, \( G_{V_i} \), an associated probability matrix/vector, \( E(v) \), and an effective virtual payoff matrix, \( G_{ev} \), where \( G_{ev} = E(v) \cdot G_n \).

3. Methodology
Here the theory of virtual games is used as the mathematical basis to model user interface designs from rival software firms. The theoretical foundations are thus based on the variant 2 game model of the theory of virtual games introduced by Nwobi-Okoye (2009).

The game model used in the analysis in this work is based on the following characteristics and assumptions:

1. The game corresponds to model/variant 2 of the game developed by Nwobi-Okoye (2009).
2. An infinite population source (Asmussen, 2003; Hamdy, 2004) which represents possible buyers of the software is assumed.
3. Purchased products are not returned, hence no possibility of moving from a higher state to a lower state.

3.1. Mathematical analysis
Here it is assumed that two competitors try to out-design each other with sophisticated software user interface. The software users could queue behind (buy) any of the software designs they found with better flow characteristics features. Such features could be features designed to induce waiting, elicit high concentration from users, make users have subjective sense of time etc. The entire mathematical analysis is entirely based on the model developed by Nwobi-Okoye (2009).

Since the game corresponds to variant 2 as has earlier been mentioned, the applicable equations for modeling the game theoretic aspect of user interfaces are:

\[
EG = P \cdot G 
\]

(1)

where \( EG, P \) and \( G \) are matrices.

The dot operator in Equation (1) carries out the operation of multiplying each payoff in the payoff matrix (\( G \)) by its associated probability of occurrence which corresponds to each element in the matrix (\( P \)).

\[
GT = G - R 
\]

(2)

Here \( R \) is the payoff reduction matrix.

Each element of \( R \) represents the extra cost incurred due to the incorporation of flow features such that for the elements of \( GT \) to be obtained the following mathematical expressions are used:

\[
GT_{IXX} = G_{IXX} - R_X 
\]

(3)

\[
GT_{IYV} = G_{IYV} - R_Y 
\]

(4)

Here \( G_{IXX} \) represents elements of player \( X \) in \( G \), \( R_X \) represents elements of player \( X \) in \( R \).
The grand payoff matrix, $G_G$, is shown in Figure 1.

Here

$G_G_{iX, jX} = \text{the cumulative payoff for player } X \text{ in the effective payoff matrix, } E_G \text{, when } X \text{ uses strategy } i - 1$.

$G_G_{iY, jY} = \text{the cumulative payoff for player } Y \text{ in the effective payoff matrix, } E_G \text{, when } Y \text{ uses strategy } j - 1$.

$E_G = PT \cdot G$  \hspace{1cm} (5)

$E_G = N \times \text{uni2211}\cdot\text{s1}$

$G_T = N \times \text{uni2211}\cdot\text{s1}$

$E_G = N \times \text{uni2211}\cdot\text{s1}$

$G_T = N \times \text{uni2211}\cdot\text{s1}$

$N = \text{maximum value of strategies i.e. possible number of software user interface design schemes.}$

4. Results from case studies

4.1. Application case study 1: Desktop application

4.1.1. Definitions

In this situation the software being evaluated are desktop applications. The underlying fact here is that the software inventory builds up and its depletion mechanism is a Markov process i.e. it follows a Markov chain (see Ching & Ng, 2006). The appropriate class of virtual games to be used for the modeling as I have said earlier is the Markov Queue Game. In this game the matrix $P$ in Equation (1) is a Markov transition matrix whose nature is shown in Figure 2.
Assuming two competitors are denoted by $X$ and $Y$, the unified transition matrix $PU$ obtained by combining the transition matrices of players $X$ and $Y$ denoted by $Px$ and $Py$ respectively is shown in Figure 3.

**Proposition 1** Let the economic order/batch quantity for player $X$ and $Y$ be denoted by $EOQ_x$ and $EOQ_y$ respectively, where $EOQ_x > EOQ_y$. If the value of $N$ in matrix $PU$ is such that $N = EOQ_x$, it follows that $GG_{IXJX} > GG_{IYJY}$ provided the payoff reduction factor $z$ is the same and the game is not biased in favour of any player.

**Proof** If $N = EOQ_x$, it follows that $N > EOQ_x$. Since $N > EOQ_x$, the batch quantity for player $Y$ is less than the economic batch quantity. Hence, player $Y$ makes less gain than player $X$. It follows that all elements of the payoff matrix $G$ or $GT$ for competitor $X$ is greater than that for competitor $Y$ since the payoff reduction factor $z$ is the same. Since from Equation (1), $EG = P \cdot G$ and from Equation (5) $EG = PT \cdot GT$, it follows that $EG_{IXX} > EG_{IYX}$ But from Equations (6) and (7) $GG_{IXX} = \sum_{i=1}^{N} \sum_{j=1}^{N} EG_{IXX}$ and $GG_{IYX} = \sum_{i=1}^{N} \sum_{j=1}^{N} EG_{IYX}$. Hence, $GG_{IXX} > GG_{IYX}$ provided the payoff reduction factor $z$ is the same and the game is not biased in favour of any player.

**Proposition 2** Let $GG_{IXnJXn}$ represent elements of competitor $X$ of the matrix $GG$ at period $n$ and $GG_{IXn+1JXn+1}$ at period $n + 1$.

It follows that $GG_{IXnJXn} > GG_{IXn+1JXn+1}$ provided the equilibrium of the transition matrix $P$ is not reached at period $n$.

**COROLLARY** Similarly, $GG_{IXnJXn} = GG_{IXn+1JXn+1}$ provided the equilibrium of the transition matrix $P$ is reached at period $n$.

**Proof** It can easily be shown that for the transition matrix $P$, that

$$\sum_{n=1}^{N} p_{ii}^{(n)} \neq \infty \text{ Provided } i \neq N$$

Therefore, each state is transient except $N$. Since state $N$ is the absorbing state and transition matrix $P$ is reducible; hence, at equilibrium the probability of staying at state $N$ is one, while the probability of staying in any other state is zero. But from Equations (6) and (7) $GG_{IXnJXn} = \sum_{i=1}^{N} \sum_{j=1}^{N} EG_{IXnJXn}$ and

$$GG_{IXn+1JXn+1} = \sum_{i=1}^{N} \sum_{j=1}^{N} EG_{IXn+1JXn+1}.$$ 

It follows that provided the equilibrium of the transition matrix $P$ is not reached, $GG_{IXnJXn} < GG_{IXn+1JXn+1}$.

**Proposition 3** For a given payoff reduction factor $z$, if strategy $n$ is optimal i.e. an equilibrium at period $n$, it may not necessarily be optimal at period $n + 1$ provided the equilibrium state of the transition matrix $P$ is not reached at period $n$ and the game is not biased in favour of any player.

**COROLLARY** For a given payoff reduction factor $z$, if strategy $n$ is optimal i.e. an equilibrium at period $n$, it is also optimal at period $n + 1$ provided the equilibrium state of the transition matrix $P$ is reached at period $n$ and the game is not biased in favour of any player.
Proof  Assuming some values of payoff reduction factor $z$ denoted by $z_1$ and $z_2$. Let $GG_{ix/nz_1}$ represent the gain for competitor $X$ while playing the optimal strategy $n$ at period $n$ when the payoff reduction factor is $z_1$ and $GG_{ix/nz_2}$ represent the gain for competitor $X$ while playing the optimal strategy $n$ at period $n + 1$ when the payoff reduction factor is $z_2$. Since $z_1 = z_2$ and from proposition 1 it is shown that at equilibrium $GG_{ix/nz_1} = GG_{ix/nz_2}$, it follows that for a given payoff reduction factor $z$, if strategy $n$ is optimal i.e. an equilibrium at period $n$, it is also optimal at period $n + 1$ provided the equilibrium state of the transition matrix $P$ is reached at period $n$ and the game is not biased in favour of any player. □

4.1.2. Problem statement

Let us assume that the profit made from selling each copy of software is $20. Assuming that adding a flow characteristics feature costs either $1 or $5 which reduces the profit $19 or $15. If we assume an infinite population source and that a batch of two pieces is produced at a time and put out for sale.

The mathematical model used in this case corresponds to variant 1 games as mentioned above. Here it is assumed that the competing firms incorporate flow features in their software products in order to induce customers to buy their products.

The queuing model for the system could be represented by Figure 4.

The servers in Figure 4 are assumed to have a finite capacity of two. Hence, a maximum of two products is in stock at any given period.

The inventory model for the system could be represented by Figure 5 below.

As shown in the inventory model in Figure 5, there is instantaneous build up of inventory from zero (0) to economic order quantity (EOQ). The depletion may be instantaneous or not.

![Figure 4. Queuing model for the system.](image-url)

**Notes:** Here $Q_a$ = Customer for firm $a$; $Q_b$ = Customer for firm $b$.

![Figure 5. Inventory model for the system.](image-url)
How could game theoretic modeling be used to determine the equilibrium point assuming two vendors make the same software and could either put flow features or not?

This problem could be solved by assuming a 2 strategy, 2-persons game which will be evaluated shortly.

4.1.3. Evaluation of the 2 strategy, 2-persons game
Typical transition matrices for the two competitors/vendors X and Y are shown below:

\[ P_Y = \begin{bmatrix}
0 & 0.20 & 0.60 & 0.20 \\
1 & 0.00 & 0.40 & 0.60 \\
2 & 0.00 & 0.00 & 1.00 \\
\end{bmatrix} \]

Next state

\[ P_X = \begin{bmatrix}
0 & 0.20 & 0.60 & 0.20 \\
1 & 0.00 & 0.40 & 0.60 \\
2 & 0.00 & 0.00 & 1.00 \\
\end{bmatrix} \]

For the above transition matrices, it is assumed that the probability of moving from a higher state to a lower state is zero (0) since the software is produced and sold in batches, and sold goods are not returned.

If the game is biased in favour of X, \( P_x \) looks like this:

\[ P_{B_X} = \begin{bmatrix}
0 & 0.10 & 0.70 & 0.20 \\
1 & 0.00 & 0.30 & 0.70 \\
2 & 0.00 & 0.00 & 1.00 \\
\end{bmatrix} \]

A unified transition matrix, \( P \), of the game produced by combining \( P_x \) and \( P_y \) is shown below:

\[ P_0 = \begin{bmatrix}
0 & 0.20 & 0.20 & 0.60 & 0.60 & 0.20 & 0.20 \\
1 & 0.00 & 0.00 & 0.40 & 0.40 & 0.60 & 0.60 \\
2 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 1.00 \\
\end{bmatrix} \]

A typical payoff matrix \( G_0 \) for the game is shown below:

\[ G_0 = \begin{bmatrix}
0 \quad 0 \quad 20,20 \quad 40,40 \\
1 \quad 0,0 \quad 0,0 \quad 20,20 \\
2 \quad 0,0 \quad 0,0 \quad 0,0 \\
\end{bmatrix} \]
The effective payoff matrix for the game, $E_{G_0}$, is given by:

$$E_{G_0} = G_0 \cdot P_0$$

$$
\begin{bmatrix}
0 & 1 & 2 \\
0 & 0.0,0.0 & 12.0,12.0 & 8.0,8.0 \\
1 & 0.0,0.0 & 0.0,0.0 & 12.0,12.0 \\
2 & 0.0,0.0 & 0.0,0.0 & 0.0,0.0
\end{bmatrix}
$$

Possible total payoff for player $X$ = 32.0

Possible total payoff for player $Y$ = 32.0

The transition and payoff matrices above represent the natural state of the game i.e. when virtual strategies are not used.

Since introducing flow features is a virtual reality, introducing it as a strategy would, as expected, improve the payoff of the competitors using the strategy (Nwobi-Okoye, 2009). Hence, a unified transition matrix of the game produced by combining matrix $P_X$ and $P_Y$ is assumed to change to $P_T$ as shown below, in accordance with virtual strategy theorem (Nwobi-Okoye, 2010a).

$$P_T = \begin{bmatrix}
0.0,0.0 & 0.7,0.0 & 0.3,0.0 \\
0.0,0.7 & 0.2,0.2 & 0.8,0.0 \\
0.0,0.3 & 0.0,0.8 & 1.0,1.0
\end{bmatrix}$$

For the payoff matrix $G_0$ above, if strategy 1 which is virtual strategy is used and $R = 1.0$, we have from Equation (2):

$$G_T = \begin{bmatrix}
-1.0, -1.0 & 19.0,19.0 & 38.0,38.0 \\
0.0,0.0 & 0.0,0.0 & 19.0,19.0 \\
0.0,0.0 & 0.0,0.0 & 0.0,0.0
\end{bmatrix}$$

$$P_T = \begin{bmatrix}
0.0,0.0 & 0.7,0.0 & 0.3,0.0 \\
0.0,0.7 & 0.2,0.2 & 0.8,0.0 \\
0.0,0.3 & 0.0,0.8 & 1.0,1.0
\end{bmatrix}$$

The effective payoff matrix, $E_{G_1}$, is given by:

$$E_{G_1} = G_T \cdot P_T$$

$$
\begin{bmatrix}
0.0,0.0 & 13.3,13.3 & 11.4,11.4 \\
0.0,0.0 & -0.2,-0.2 & 15.2,15.2 \\
0.0,0.0 & 0.0,0.0 & 0.0,0.0
\end{bmatrix}
$$

Total payoff for player $X$ = 37.7

Total payoff for player $Y$ = 37.7
The grand payoff matrix, GG for $R = 1.0$ is given by:

$$
GG = \begin{bmatrix}
0 & 32.0, 32.0 \\
32.0, 37.7 & 1
\end{bmatrix}
$$

A look at the payoff matrix, GG, above shows that Nash equilibrium point corresponds to strategy 1, 1.

For the payoff matrix $G_0$ above, if strategy 1 which is virtual strategy is used and $R = 5.0$, we have from Equation (2):

$$
GT = \begin{bmatrix}
-5.0, -5.0 & 15.0, 15.0 & 30.0, 30.0 \\
0.0, 0.0 & 0.0, 0.0 & 15.0, 15.0 \\
0.0, 0.0 & 0.0, 0.0 & 0.0, 0.0
\end{bmatrix}
$$

$$
PT = \begin{bmatrix}
0.0, 0.0 & 0.7, 0.0 & 0.3, 0.0 \\
0.0, 0.7 & 0.2, 0.2 & 0.8, 0.0 \\
0.0, 0.3 & 0.0, 0.8 & 1.0, 1.0
\end{bmatrix}
$$

The effective payoff matrix, $EG_0$, is given by:

$$
EG = GT \cdot P
$$

$$
EG = \begin{bmatrix}
0.0, 0.0 & 10.5, 10.5 & 9.0, 9.0 \\
0.0, 0.0 & -0.1, -0.1 & 12.0, 12.0 \\
0.0, 0.0 & 0.0, 0.0 & 0.0, 0.0
\end{bmatrix}
$$

Total payoff for player $X = 37.7$

Total payoff for player $Y = 37.7$

The grand payoff matrix, GG for $R = 5.0$ is given by:

$$
GG = \begin{bmatrix}
0 & 32.0, 32.0 \\
32.0, 30.5 & 1
\end{bmatrix}
$$

A look at the payoff matrix, GG, above shows that Nash equilibrium (Nash, 1950) point corresponds to strategy 0, 0.

4.2. Application case study 2: Web applications

In this case websites or web applications are being evaluated. The evaluation could equally be for desktop applications being purchased online or downloaded from websites. The appropriate class of virtual games to be used for the modelling is the slightly modified version of the Markov Queue Game. In this game the matrix $P$ is a Markov transition matrix whose nature is shown in Figure 6. In
matrix $P$, $P_{iX,jX}$ represents elements for competitor $x$, while $P_{iY,jY}$ represents elements for competitor $y$ assuming a 2-persons game. The elements of $P$ may or may not be as shown in Figure 2.

**Proposition 4** Consider a set of web pages given by:

$$W = \{1, 2, 3 \ldots n\}$$

Let $R_i$ be the rank of web page $i$.

Consider a scenario or situation where $i$ became incorporated with flow features. Let $R_j$ be the rank of $i$ after this change. It follows that

$$R_j > R_i$$

**Proof** Consider the matrix $P$. let the set $F$ given by:

$$F = \{F_1, F_2, F_3 \ldots F_n\}$$

$F$ represents the set of probabilities of visit to web page $i$. Let sum of these probabilities be given by:

$$\text{Sum1} = \sum_{j=1}^{n} F_i$$

Consider the matrix $Pv$ after a virtual strategy (flow features) was introduced. Let the set $V$ be given by:

$$V = \{V_1, V_2, V_3 \ldots V_n\}$$

$V$ represents the set of probabilities of visit to web page $j$. Let sum of these probabilities be given by:

$$\text{Sum2} = \sum_{j=1}^{n} V_j$$

It follows from virtual strategy theorem (Nwobi-Okoye, 2010a) that $\text{Sum2} > \text{Sum1}$, hence $R_j > R_i$. □

**Proposition 5** Incorporating flow features in web page design would always be the optimum strategy provided no counter strategy is employed by an opponent, hence the equilibrium point in matrix GG must correspond to a position where $i \neq 0$ or $j \neq 0$.

**Proof** From proposition 1, $R_j > R_i$. It follows from matrix GG that for any competitor $i$ or $j$,

$$GG_{iX,iX} > GG_{iX}$$

and
providing $i \neq 0$ or $j \neq 0$, hence flow would always be the optimum strategy.

4.2.1. Problem statement
Consider a Markov process which has state space $S$ such that:

$$S = \{0, 1, 2 \ldots n\}$$

The states are:

$$M = \{0, 1, 2, 3 \ldots n\}$$

The states represent web pages. State 0 assumes that no page is visited, state 1 assumes web page 1 is visited, state 2 assumes web page 2 is visited, state 3 assumes web page 3 is visited etc.

The transition diagram for the system is shown in Figure 7;

The transition matrix is shown below:

For the transition matrix $P_0$ above, at equilibrium the proportion of visits to the web pages is shown in matrix $V_e$ below:

$$V_e = \begin{bmatrix} 0.0625 & 0.3750 & 0.3702 \end{bmatrix}$$

Hence, the rank for the web pages is 0.0625, 0.3750 and 0.3702 for web pages 1, 2 and 3 respectively.

For the above transition matrix $P_e$, it is assumed that the probability of moving from a higher state to a lower state is not zero (0) since a web surfer or user could move from one web page to another or could stop surfing or browsing entirely. This is a slight change/modification from the model of the typical Markov queue game developed by Nwobi-Okoye (2009) where it is assumed that the probability of moving/transiting from a higher state to a lower state is zero (0).
The transition matrix $P_x$ above represents the natural state of the game. Assuming web page 1 is incorporated with flow features, the strategy is a virtual strategy and from virtual strategy theorem, the attached probability matrix $P$ must change due to increased traffic to web page 1. Hence, matrix $P$ would change to matrix $P_T$ as shown below:

$$P_0 = \begin{bmatrix}
0 & 0.20 & 0.50 & 0.30 \\
1 & 0.10 & 0.10 & 0.30 & 0.50 \\
2 & 0.20 & 0.00 & 0.20 & 0.60 \\
3 & 0.30 & 0.10 & 0.50 & 0.10 \\
\end{bmatrix}$$

For the transition matrix $P_T$ above, at equilibrium the proportion of visits to the web pages is shown in matrix $V_b$ below:

$$V_b = [0.0818 \ 0.3720 \ 0.3566]$$

Hence, the rank for the web pages is 0.0818, 0.3720 and 0.3566 for web pages 1, 2 and 3 respectively.

Assuming that the payoff matrix $G_0$ is as shown below and the elements of the matrix represent the payoff per visit to each web page.

$$G_0 = \begin{bmatrix}
1 & 2 & 3 \\
200 & 500 & 300 \\
\end{bmatrix}$$

The effective payoff matrix for the game, $E G_0$, is given by:

$$E G_0 = G_0 \cdot V_a$$

Here $V_a \equiv P_0$ in Equation (1).

$$E G_0 = \begin{bmatrix}
12.50 & 187.50 & 111.06 \\
\end{bmatrix}$$

Possible total payoff for player 1 = 12.50
Possible total payoff for player 2 = 187.50
Possible total payoff for player 3 = 111.06

The transition and payoff matrices $P_0$ and $G_0$ above represent the natural state of the game.

Assuming that the payoff matrix after the introduction of the flow features page is represented by $G_T$ is as shown below and the elements represent the payoff or possible gain per visit to each web page.

$$G_T = \begin{bmatrix}
1 & 2 & 3 \\
200 & 500 & 300 \\
\end{bmatrix}$$

Note that $G_0$ and $G_T$ are equivalent because it is assumed that the cost of introducing the flow features on the web page is negligible for the purposes of the analysis here.

The effective payoff matrix, $E G_T$, is given by:
\[ EG_1 = GT \cdot V_b \]

Here \( V_b \equiv PT \) in Equation (5).

\[ EG_1 = \begin{bmatrix} 16.36 & 186.00 & 106.98 \end{bmatrix} \]

Possible total payoff for player 1 = 16.36

Possible total payoff for player 2 = 186.00

Possible total payoff for player 3 = 106.98

Excluding players 2 and 3, let us consider the one person game represented by GG.

\[ GG = \begin{bmatrix} 0 & 12.50 & 16.36 \end{bmatrix} \]

A look at the payoff matrix, GG, above shows that the optimum strategy for player 1 corresponds to strategy 1, which is the virtual strategy, hence, Nash equilibrium point corresponds to strategy 1.

Let \( U_f(w) \) be the utility obtained by a firm from user interface \( i \). In general, the equilibrium/optimal user interface is given by:

\[ \max_{R_i} U_f(w)R_i \]

where \( R_i \) is the rank of user interface \( i \).

5. Results and discussion

From the evaluation above it could be deduced that the application virtual strategies (flow features), increased the possibility of either vendor selling its own software. Propositions 1, 3, 4 and 5, buttress this fact. But application of virtual strategy comes at a price. Hence, at a certain cost the virtual strategy becomes optimal for either player. But as cost increases, non virtual strategy becomes optimal. Hence, there must always be a complete economic analysis before applying virtual strategies in Markov Queue Games. This fact is exemplified by android devices where Google licenses the android software free to manufacturers. Hence, even though the Iphone from apple was introduced in 2007 much earlier than Android which was introduced in 2008, the zero cost of licensing Android, which has comparable user experience to Iphone, to manufacturers contributed to Android competing favourably with Iphone. Thus at the moment android devices constitute 85percent of the smart phone market in the first quarter of 2017 (International Data Cooperation (IDC), 2017). Similarly, MySpace used to be the top social networking site, but Facebook with superior user interface design quickly overtook it to be the largest social networking site with 2Billion active users. MySpace currently has 37million active users.

The overall effect of incorporating flow features in software products is that it could improve the fortunes of firms. This could be explained from the fact that incorporating flow features has the tendency to make more customers buy the products, and an infinite population source is assumed. This phenomenon is akin to advertising which could make more buyers to buy competing products, thereby improving the fortunes of the competing firms (Nwobi-Okoye, 2010b). This fact is buttressed by the statistics which shows that increasing number of people are migrating from using feature phones (dumb phones) to smart phones due to its superior user experience.

From proposition 1 it is clear that any batch size that is not equal to the economic batch size will not result to optimal gain for any player as long as the conditions remain the same for the competing firms. Hence, the economic batch size must be determined by the competitors, and the maximum quantity of products in stock must be the economic batch size.
From the analysis above, it is obvious that making a web page have flow features would improve/increase the page rank. Increasing the page rank results to increased payoff for the player using flow features, a virtual strategy. Hence, flow would always be optimum for players using such strategies. Proposition 4 gives the scientific bases why websites with very high usability and overall user experience will be ranked higher in search engine results than poorly designed websites with poor user interfaces.

The consequence of proposition 2, which applies to both desktop and web applications, is that a player that starts the game earlier than his opponent is more likely to win. Hence any firm that enters the market first is most likely to make more profits provided all conditions remain the same for the competing firms. Thus even though windows phone 7 devices are comparable to Iphone and Android devices in user experience, they are struggling to compete with them because these devices were introduced to the market much earlier than windows phone 7 devices. Secondly, windows phone 7 (seven) software is not free to hardware manufacturers which compound the misery of windows phone 7. Hence, it should be more favourable to Microsoft if it licenses windows phone 7 (seven) software free to phone manufacturers in order to increase the payoff to manufacturers of windows phone 7 devices. By so doing they would be better assured of their survival in the competitive world of smart phones. Similarly, it could be more favourable to Research in Motion (RIM), the makers of blackberry devices to adopt Google's Android software for their phones in order to ensure their survival. In web applications, Google Plus, the social network from Google, though it has almost similar user experience as Facebook has 170 million users way behind Facebook. This is because Google Plus started in 2011 whereas Facebook started in 2004.

In evaluating user interfaces using game theory it is necessary to use prototyping techniques. Hence, beta copies of the interface (high fidelity prototypes) are made available to users in order to gather data for game theoretic analysis. The assistance of search engines could be used in the prototype evaluations, the foundations of this is based on game theory, as shown in this work. Hence, game theoretic approach could be the best method for software performance evaluation.

6. Conclusions
Adding flow characteristics features enhances user experience in software which makes time seem to fly. This is a powerful virtual strategy which should be exploited to the maximum by software developers and designers. The mathematical/analytical basis for its successful exploitation has been laid down in this work. We have discussed extensively the practical consequences of this research as typified by the outcomes of the fierce competition in the smart phone world.

Furthermore, this work should be of immense importance to software designers and companies. This is because it is only by seeing software design as a game with several competitors will they be able to survive in the modern competitive software world. Telecommunication companies and regulators have learnt this fact and are increasingly resorting to game theory for bidding and auctioning band widths respectively (Binmore, 2008; Fudenberg & Tirole, 1991).

The analysis done in this work will be very useful to game theorists, management scientists, systems scientists and engineers, operations researchers/industrial engineers who consult on software ergonomics and design, as well as strategic management.


