



(19) **United States**

(12) **Patent Application Publication**  
Lee

(10) **Pub. No.: US 2007/0093285 A1**

(43) **Pub. Date: Apr. 26, 2007**

(54) **MULTI-PATTERN ONLINE BINGO**

**Publication Classification**

(76) Inventor: **Jordan Lee**, Richmond Hill (CA)

(51) **Int. Cl.**  
*A63F 9/24* (2006.01)

Correspondence Address:  
**RENNER, KENNER, GREIVE, BOBAK,  
TAYLOR & WEBER  
FIRST NATIONAL TOWER FOURTH FLOOR  
106 S. MAIN STREET  
AKRON, OH 44308 (US)**

(52) **U.S. Cl.** ..... 463/19

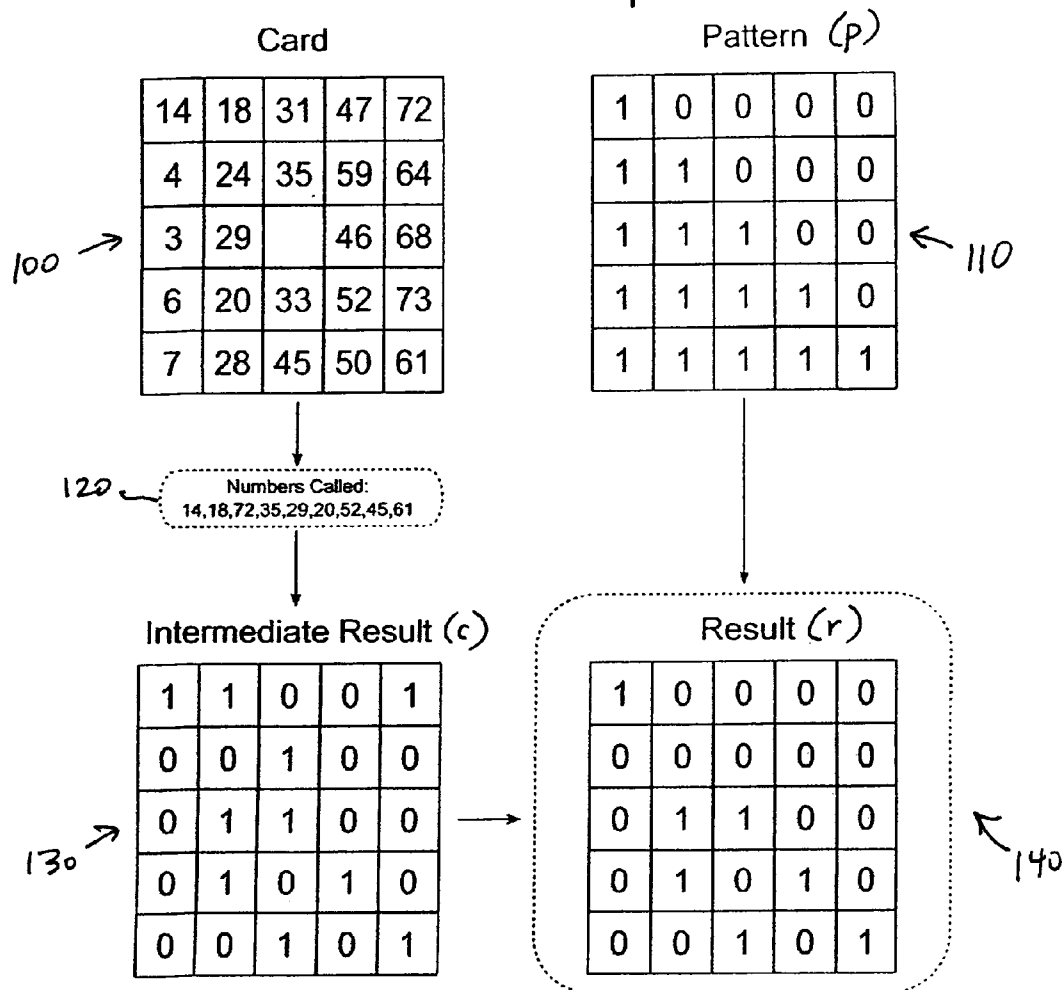
(57) **ABSTRACT**

A multi-pattern bingo game is provided. Using matrix multiplication, a selected bingo card is compared against multiple patterns that are played for simultaneously in a bingo game. If a pattern match is detected, the card is a winning card. In an online version of the game, winning cards may be determined and the information forwarded to individual player computers for simulation of the bingo game.

(21) Appl. No.: **11/258,754**

(22) Filed: **Oct. 25, 2005**

### 75 ball example



# 75 ball example

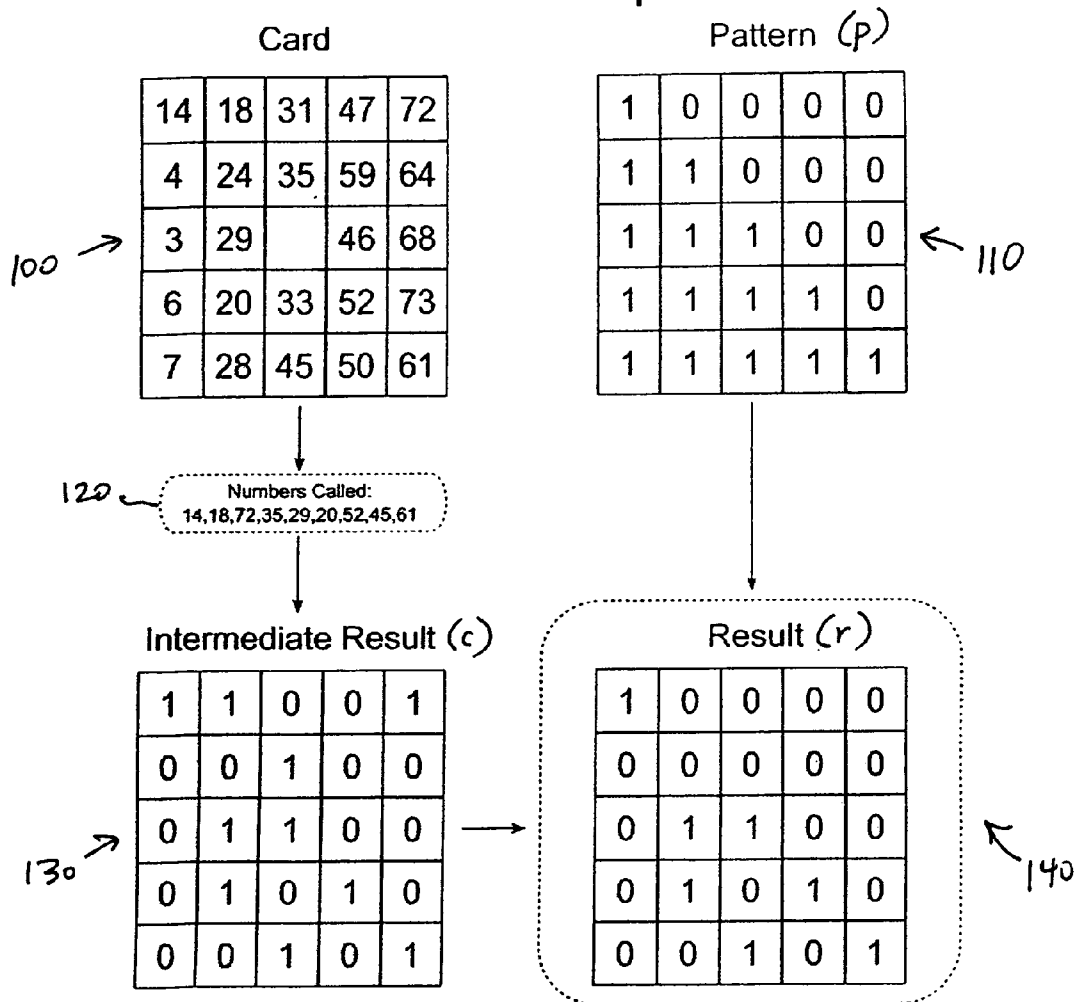


Fig. 1

# 90 ball example

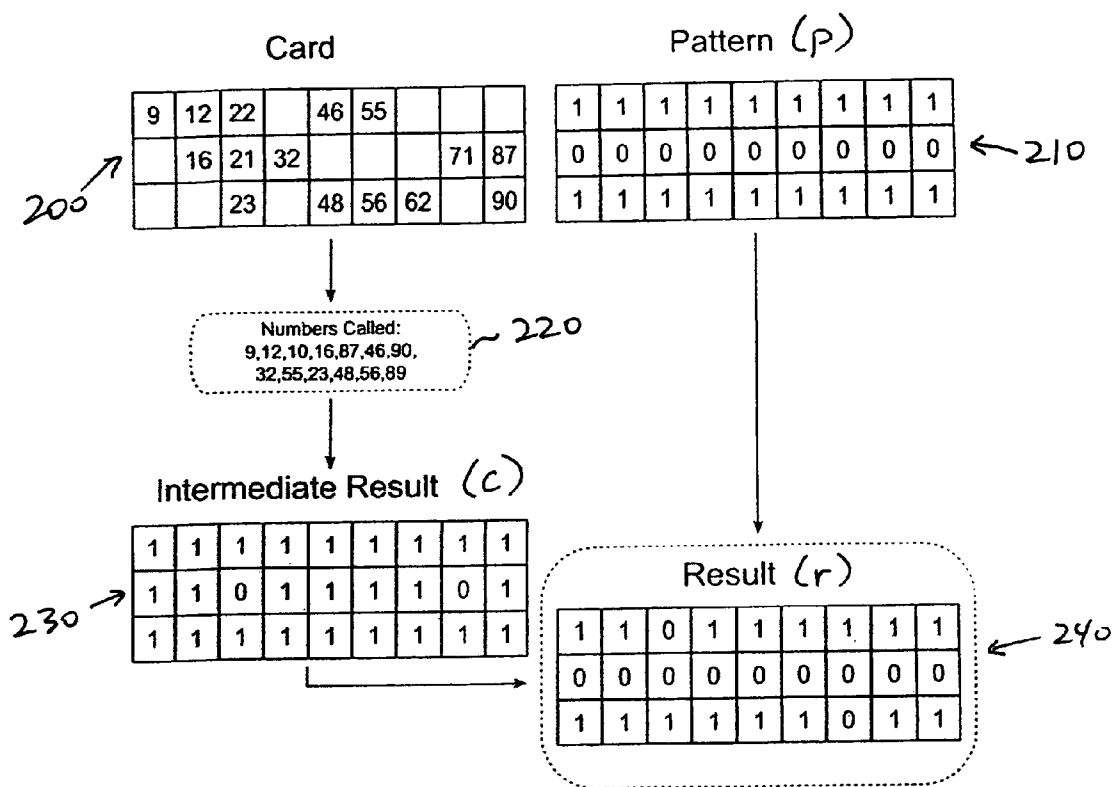


Fig. 2

# calculating remaining squares

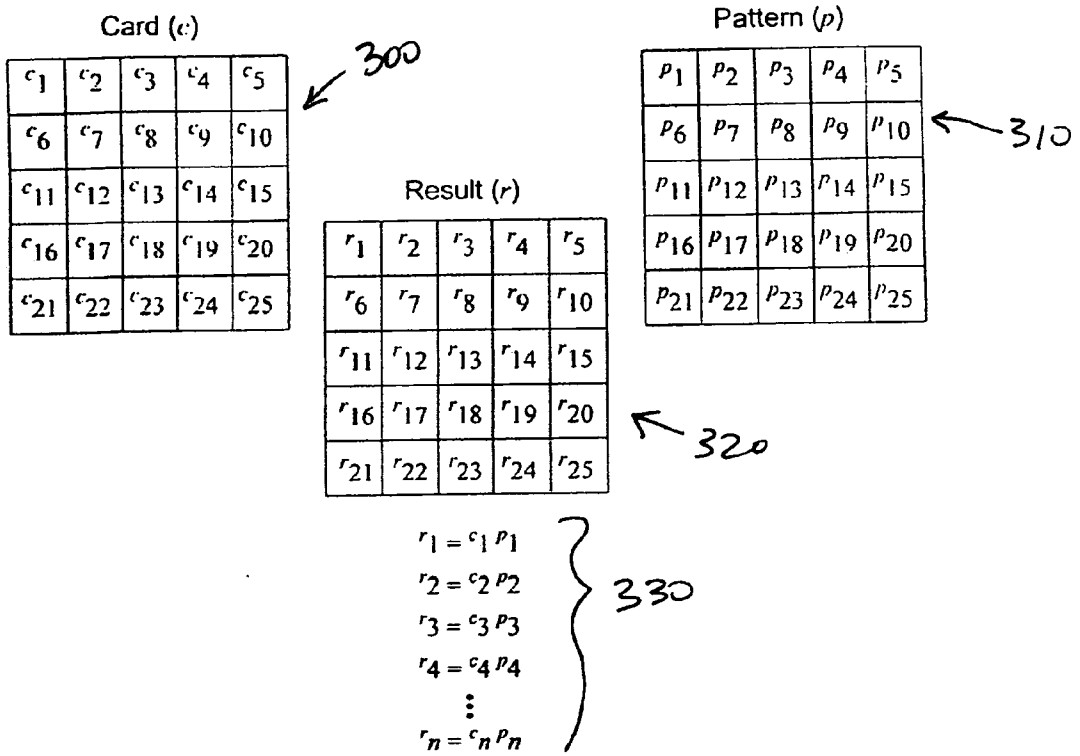
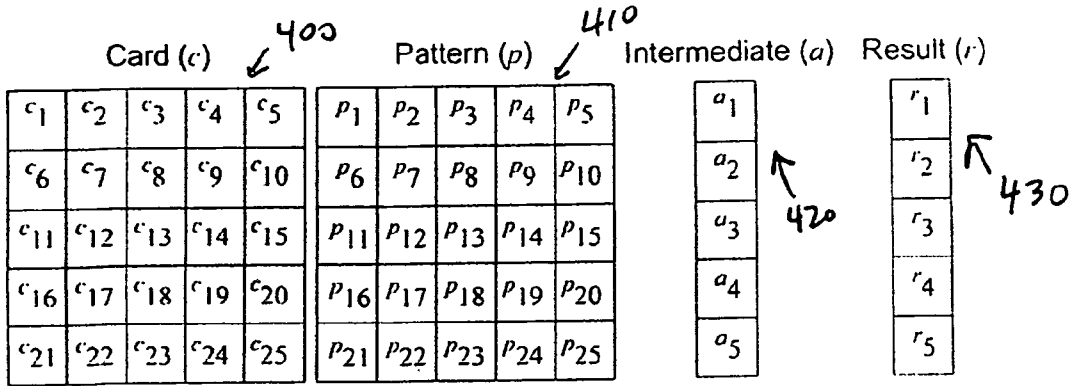


Fig. 3

# calculating "number to go"



$$\begin{aligned}
 a_1 &= p_1 + p_2 + p_3 + p_4 + p_5 \\
 a_2 &= p_6 + p_7 + p_8 + p_9 + p_{10} \\
 a_3 &= p_{11} + p_{12} + p_{13} + p_{14} + p_{15} \\
 a_4 &= p_{16} + p_{17} + p_{18} + p_{19} + p_{20} \\
 a_5 &= p_{21} + p_{22} + p_{23} + p_{24} + p_{25}
 \end{aligned}$$

} 440

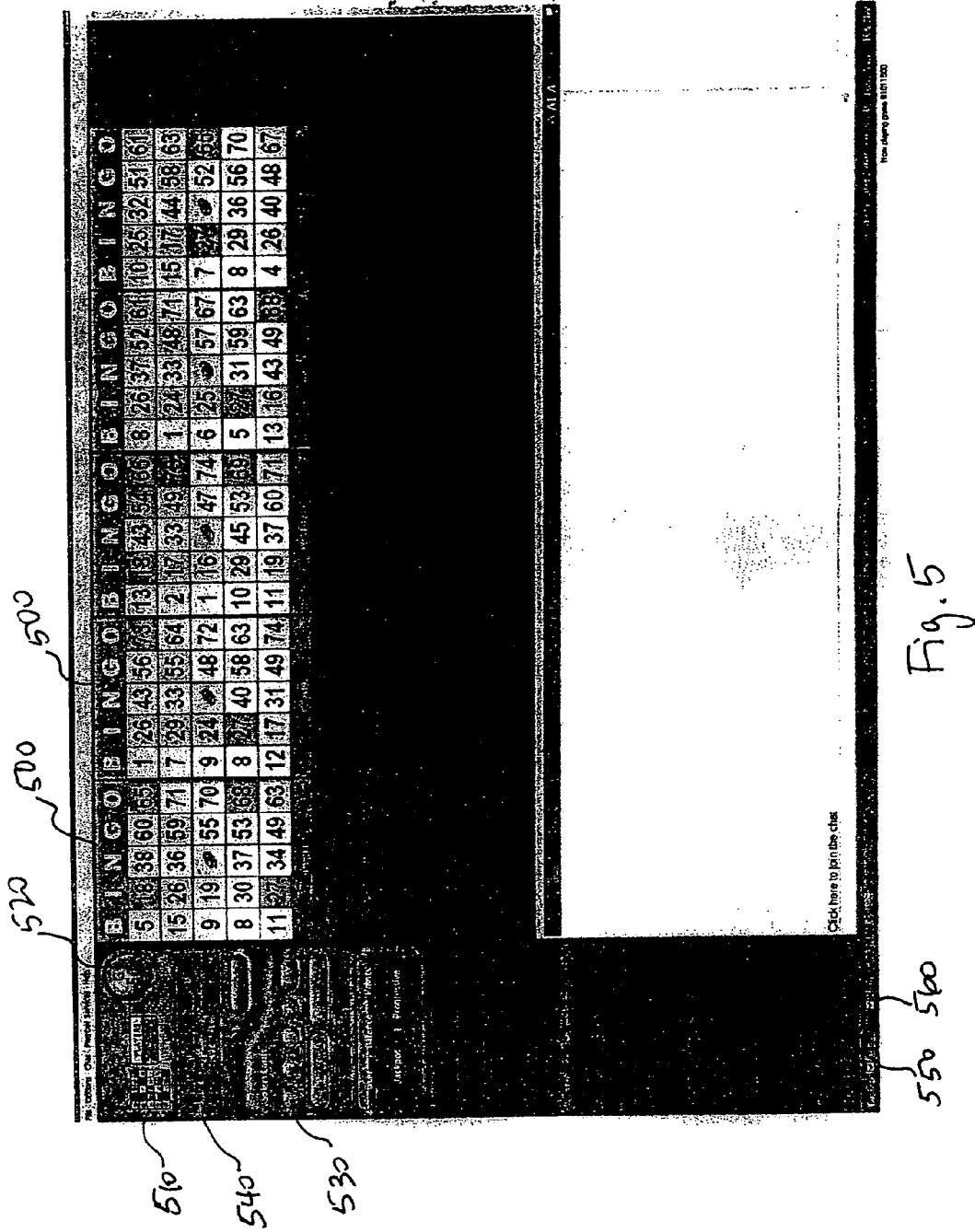
$$\begin{aligned}
 r_1 &= c_1 p_1 + c_2 p_2 + c_3 p_3 + c_4 p_4 + c_5 p_5 \\
 r_2 &= c_6 p_6 + c_7 p_7 + c_8 p_8 + c_9 p_9 + c_{10} p_{10} \\
 r_3 &= c_{11} p_{11} + c_{12} p_{12} + c_{13} p_{13} + c_{14} p_{14} + c_{15} p_{15} \\
 r_4 &= c_{16} p_{16} + c_{17} p_{17} + c_{18} p_{18} + c_{19} p_{19} + c_{20} p_{20} \\
 r_5 &= c_{21} p_{21} + c_{22} p_{22} + c_{23} p_{23} + c_{24} p_{24} + c_{25} p_{25}
 \end{aligned}$$

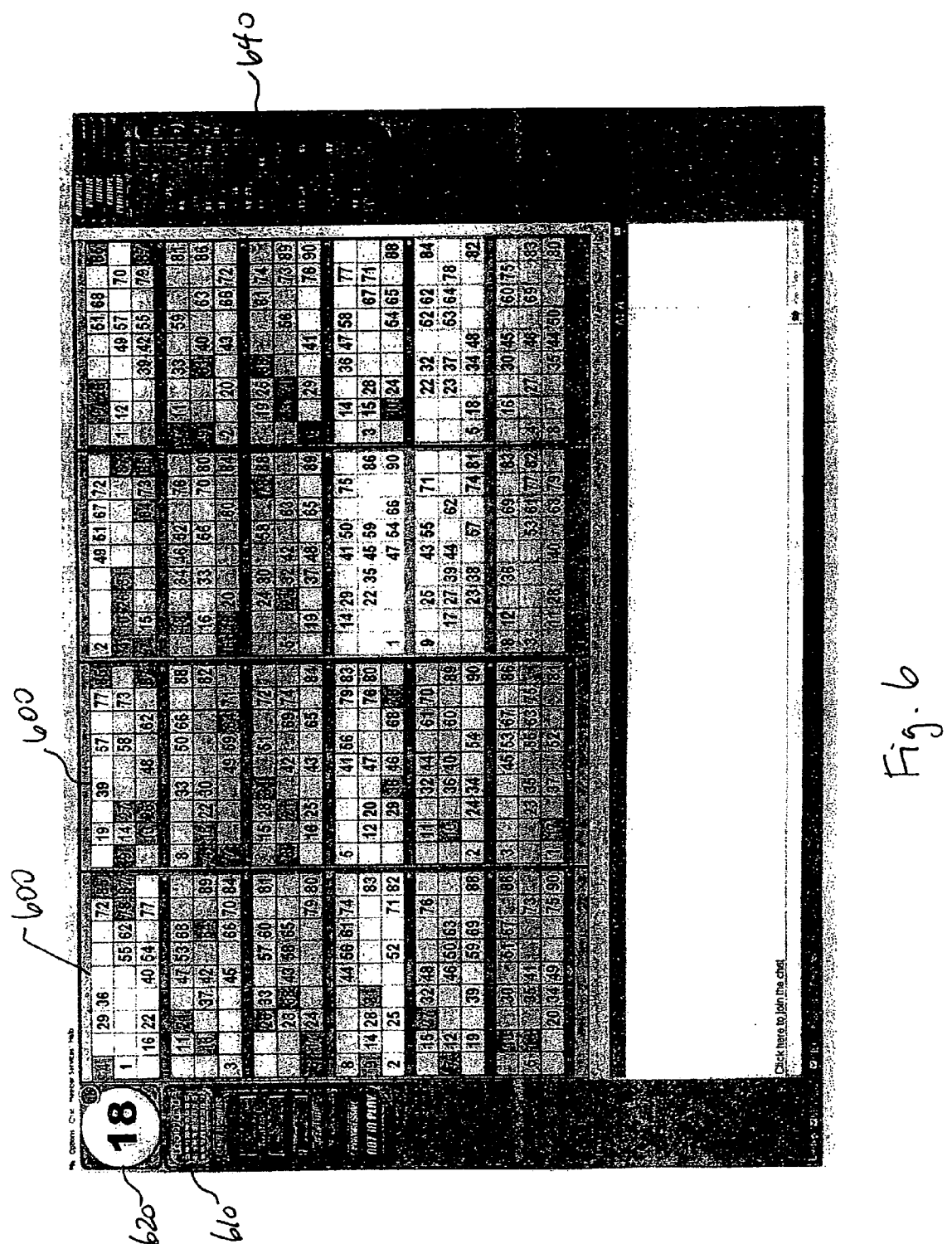
} 450

$$n = (a_1 + a_2 + a_3 + a_4 + a_5) - (r_1 + r_2 + r_3 + r_4 + r_5)$$

} 460

Fig. 4





**MULTI-PATTERN ONLINE BINGO**

**FIELD OF THE INVENTION**

[0001] The invention relates to bingo games, and more particularly to bingo games played on a computer.

**BACKGROUND OF THE INVENTION**

[0002] Bingo has been played for many years and its popularity continues to grow. In its simplest form, the game employs paper cards containing numbers in a grid formation. The cards contain numbers selected at random from a set of possible numbers (or letter/number codes). Cards are distributed to players (for free or for money) and the game play begins. An operator announces or calls numbers drawn at random. Each player marks his/her paper card as a number is matched. If the daubed numbers on the card match a pattern that has been set as the played-for pattern, then the player yells "bingo".

[0003] In some instances, the "bingo" leads to a new round and new cards must be selected/purchased. However, the play in a single round may also continue to a second or further "bingo", such as to make a pattern that is a compound pattern of the first pattern. For example, the first played-for pattern might be a single outside line, while the second played-for pattern might be a box (□), and a further played-for pattern might be a "blackout" (i.e. every space on the card grid filled in). This type of sequential play is sometimes called "progressive bingo".

[0004] Various forms of computerized and online bingo have been proposed and realized. The systems, in one form or another, all attempt to replicate the in-person experience of playing bingo in a bingo hall. However, the systems may be clunky and unappealing precisely because of the attempt to latch on to the known methods of real world bingo playing. As processing power becomes increasingly available and inexpensive, it would be desirable to use this to add greater feature-richness and excitement to the bingo game. One such improvement would be to change one of the fundamentals of the bingo game—that the players play for one and only one pattern at a time.

**SUMMARY OF THE INVENTION**

[0005] The present invention is directed at providing a true multiple pattern capability in computerized and online bingo. Unlike in traditional paper-based bingo, the multiple patterns are not played for sequentially, but simultaneously. In the paper-based bingo world, this would be confusing and difficult to implement as players would have difficulty identifying when to shout "bingo". However, the applicants have made the surprising discovery of a way to implement multi-pattern play in a computerized/online bingo game. This has been found to increase the excitement and perception of "unpredictability" in a game as players can win on any of several patterns. Each "bingo" also comes up faster, which keeps the excitement high, without the need to increase the speed of the calls. Depending on the set-up of the game, the multiple pattern play can also increase the card turnover, which may benefit the operator of the game who can therefore sell more cards in a given span of time.

[0006] One of the difficult aspects of providing multiple pattern play is the determination of winning cards. An

efficient method is needed for finding winning cards as each pattern must be compared to each card at each call. Furthermore, the method must have a way of tracking the number of calls to reach the first, second and subsequent winning cards. The applicants have found a way to do this using matrices for an elegant and efficient comparison process.

[0007] Adding to the efficiency (for an online environment particularly) the applicants' method can perform all of the calculations in advance, sending results to the client computers, thus making the "game" seen by the players effectively a re-creation or simulation. Of course, it will be understood that the calculations can be performed live, i.e. concurrently with the game play as seen by client computers.

[0008] According to a first aspect of the invention, a method is provided for determining a winning card in a multi-pattern computerized bingo game. The method comprises the following steps:

[0009] a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;

[0010] b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;

[0011] c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;

[0012] d. calling the call numbers in the sequence to construct, for each card, a daub matrix of grid positions corresponding to any card numbers which match the called number, and, for each called number:

[0013] (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and

[0014] (ii) determining from each product matrix whether the card matches the pattern; and

[0015] e. repeating steps (i) and (ii) at least until such time as a pattern match is determined for any pattern, the card being thereby determined as a winning card, and determining how many called numbers must be called to generate said winning card.

[0016] Steps (i) and (ii) may be repeated in a game until a predetermined number of winning cards is reached. Alternatively, steps (i) and (ii) may be repeated in a game until a predetermined number of called numbers is called in the sequence, regardless of the number of winning cards determined.

[0017] Each daub matrix comprises a matrix of 0s and 1s, each 1 representing a match in the grid position. The determining step involves evaluating each grid position in the product matrix and finding a pattern match if values stored in the grid positions making up the pattern are greater than zero.

[0018] In a given game, it may be set that only one winning card is determined, or the game may be continued after a winning card is determined. The number of winning



cards may be predetermined by an operator. The number of called numbers may also be predetermined by an operator.

[0019] There may be 75 call numbers in the sequence. There may be 90 call numbers in the sequence. These represent American style and European style bingo games, respectively. However, other call number configurations may also be provided.

[0020] In a preferred embodiment of the method, the bingo game is accessible to players over a distributed network (e.g. the Internet). In this embodiment, a game server is provided that is programmed to determine the winning card or cards. Client computers are also provided, and the game server (which may physically consist of one or more server machines) is in communication with the client computers. The server communicates the winning card or cards to the client computers. Each client computer is programmed for displaying a game re-creation and announcing the winning card or cards. The game re-creation is preferably interactive.

[0021] The multiple patterns may be set by an operator or by a player. These can be any configuration of markable bingo card positions—representing an identifiable shape (such as an X, an L or a line across or down) or a random configuration. Multiple patterns are played for simultaneously in a single bingo game. All cards in play in a single game are evaluated for matching the multiple patterns at the same time. In the game re-creation, each pattern may be displayed graphically for the players. There may also be a display indicating, for each card, and each of the multiple patterns, how many “numbers away” from matching the pattern the card is at a given time in the re-creation.

[0022] During the game re-creation, the player’s card may be marked automatically to show the card numbers that have been “called” in the sequence. This is known as an “auto-daub” feature. The client computer may be programmed to provide this feature as a toggle option. If auto-daub feature is off, the player may be permitted to manually mark (or daub) his/her card(s) during the calling of the numbers. If the player gets behind in marking the card(s), the program may also provide the option to “catch-up” (i.e. to automatically mark the missed numbers).

[0023] According to a second aspect of the invention, a method is provided for determining whether any winning cards have been selected in a multi-pattern computerized bingo game. The method comprises the following steps:

[0024] a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;

[0025] b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;

[0026] c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;

[0027] d. calling the call numbers in the sequence to construct, for each selected card, a daub matrix of grid

positions corresponding to any card numbers which match the called number, and, for each called number:

[0028] (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and

[0029] (ii) determining from each product matrix whether the card matches the pattern, the card being thereby determined to be a winning card; and

[0030] e. repeating step (d) until a predetermined number of call numbers have been called, and determining how many called numbers must be called to generate a winning card, if any winning cards can be determined in the predetermined number of call numbers.

[0031] According to a third aspect of the invention, a method is provided for determining a winning card in a multi-pattern online bingo game accessible to players over a distributed network, comprising programmed steps for execution on a server. The steps include:

[0032] a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;

[0033] b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;

[0034] c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;

[0035] d. calling the call numbers in the sequence to construct, for each card, a daub matrix of grid positions corresponding to any card numbers which match the called number, and, for each called number:

[0036] (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and

[0037] (ii) determining from each product matrix whether the card matches the pattern; and

[0038] e. repeating steps (i) and (ii) at least until such time as a pattern match is determined for any pattern, the card being thereby determined as a winning card, and determining how many called numbers must be called to generate said winning card,

[0039] wherein said server is programmed for communicating the winning card or cards to a plurality of client computers, each said client computer being programmed for displaying a game re-creation to announce the winning card or cards.

#### BRIEF DESCRIPTION OF THE FIGURES

[0040] FIG. 1 is a diagram of sample bingo card and flow of computation to compare the card against one sample pattern in a game (for a 75 ball example).

[0041] FIG. 2 is a diagram of sample bingo card and flow of computation to compare the card against one sample pattern in a game (for a 90 ball example).

[0042] FIG. 3 is a diagram illustrating the multiplication of card (c) by pattern (p) to generate resulting matrix (r).

[0043] FIG. 4 is a diagram illustrating the computation to determine “number away” or “number to go” between a card (c) and pattern (p).

[0044] FIG. 5 is a sample screen shot of a computerized 75 ball bingo game.

[0045] FIG. 6 is a sample screen shot of a computerized 90 ball bingo game.

#### DETAILED DESCRIPTION OF THE FIGURES

[0046] As shown in the figures, the invention uses matrices for determining whether a pattern match has been achieved.

[0047] An example of such comparison in a 75 ball bingo game is shown in FIG. 1. The card 100 is a standard 5x5 grid having card numbers between 1 and 75. The middle square, as in conventional paper-based bingo, is a “free space” without a number. The card represents a card selected (or purchased) by a player. As will be apparent to a person skilled in the art, many players will select to “play” on multiple cards in a given game. In a preferred embodiment, each player is presented with a bank of potential cards from which to select cards to be played in the course of the game. At the point of selecting cards, the player may or may not know which patterns are being played for in the game.

[0048] One of the multiple patterns played for in the sample game is shown (p) at 110. Grid spaces that must be marked to match the pattern are represented by 1’s in the pattern (p) 110, while non-pattern spaces are represented by 0’s. As illustrated at 110, the pattern compared in this case is a triangle pointed toward the lower left corner of the grid.

[0049] A call number sequence having already been determined before the card was selected, the comparison proceeds as each number in the sequence is “called” 120. (It will be understood that the “calling” of numbers is a virtual not literal process. Each number in the sequence is simply reviewed in the sequence.) As each number is called, a daub matrix (“intermediate result” (c) 130) is updated by marking with a “1” any positions on the grid where a number match has been determined (i.e. by the comparison between the 1-75 numbers on the card 100 and the call numbers 120). As yet unmarked spaces on the daub matrix 130 are represented as “0”.

[0050] To determine result matrix (r) 140, the pattern matrix (p) 110 is multiplied by the daub matrix 130. Each grid position in the result matrix 140 thus represents the product of  $c_n * p_n$  (where n is the grid position in each matrix). As can be seen from 140 in FIG. 1, a pattern match has not yet been achieved in the example. In order to match the pattern, result matrix (r) will need to have 1’s in each of the pattern spaces, according to the pattern (p).

[0051] A different example is shown in FIG. 2 for a 90 ball bingo game. The pattern (p) 210 in this case is two lines running across the top and bottom of the grid. Similar to the process described with reference to FIG. 1, the intermediate result (c) grid 230 is determined from simulating the daubing of the card 200 according to the numbers called 220. This grid 230 is then multiplied by the pattern grid 210 to

obtain result (r) 240. As shown in FIG. 2, a pattern match has not yet been determined based on the numbers called.

[0052] The use of 1’s and 0’s is particularly useful in the present matrix computation, as these numbers have particular properties in multiplication. Multiplication by 1 maintains “identity”. Therefore, multiplying 1\*1 in a grid position produces 1. Multiplication by 0 always produces 0. Therefore, multiplying either 0 or 1 by 0 produces 0.

[0053] In the present case, the result matrix (r) 140 will not display any 1’s in any non-pattern positions. It will only “credit” or acknowledge 1’s in pattern positions. Therefore, part of the elegance in the present computation is that a pattern match will always be  $r=p$ . Stray numbers will not appear in the non-pattern spaces of the result matrix (r) 140.

[0054] The calculation for remaining squares is shown as a mathematical model in FIG. 3. As shown at 330, the multiplication of each grid position in card (c) 300 (i.e. daub matrix) by each grid position in pattern (p) 310 produces result matrix (r) 320. Said in another way,  $r_n = c_n p_n$  (where n is any grid position).

[0055] FIG. 4 provides a mathematical model illustrating how the “number to go” or “number away” can be calculated. The values in the pattern matrix (p) 410 are added together across rows to obtain a one-dimensional matrix (a) 420, as illustrated at 440. The values in the daub matrix at (c) 400 are multiplied by values in the pattern matrix (p) 410 and the products added row-by-row to obtain a second one-dimensional matrix (r) 430, as illustrated at 450. To obtain the “number to go” (n), the result matrix (r) is subtracted from the intermediate matrix (a), as illustrated at 460.

[0056] In the course of the method as set out in the claims, the mathematical calculations are done for each card, for each of the multiple patterns, as each number is called. All of this is done before a player at a client computer sees the progress of the “game” in the usual sense. The game seen by the player is in effect a re-creation or simulation of the process that has already gone on—at high speed—in advance by the server. When a player sees a screen, such as shown in FIG. 5 or FIG. 6, showing his/her selected cards (500 or 600), the computer already knows which of those cards can win on which patterns and how many called numbers it will take to get there. Of course, none of this information is revealed to the player or players, so the game appears to unfold in the usual sense, with numbers being called and numbers on the cards being marked as they come up. The player can choose to mark (daub) the cards him/herself or can opt for the program to do the marking automatically (auto-daub) 550. Preferably, the screen shows a display of the numbers called so far (540 or 640) and each number is displayed as it is called (520 or 620). Various options may exist for numbers to be called in audio mode 560 as well, and the game may, for instance, provide choices of voice characters to announce the numbers. Preferably, winning cards are announced game-wide and individual players may be eligible to win prizes.

[0057] Each game plays simultaneously for multiple patterns. The patterns may be shown in a graphical display (510 or 610) that cycles through the patterns in order. There may also be an indication on the displayed cards themselves what each pattern should look like (such as by shading or coloring the squares needed for the pattern match).

[0058] Playing for multiple patterns does not have an impact on the basic odds of the game. That is, each card played by each player has an equal chance of winning. Provided that the numbers for the call sequence are selected randomly and the outcomes are maintained in secrecy until after the card selection period has closed, the fairness of the game is preserved. Not all of the possible numbers in the sequence will need to be called. The operator can control the payout by setting how many prizes he/she is willing to award. Alternatively, the operator can control the number of called numbers without knowing how many prizes will be awarded in that time.

[0059] Various hardware configurations of the system are possible. For instance, the method could be implemented using a client-server model in which a server entity is used to process the game data and then transmit the output to one or more client machines. The client-server model could also be implemented using one or more game terminals as clients, such as terminals using touch screens. The client-server could also be implemented in a casino or bingo hall environment where the game terminals are multi-function, operating the bingo game as part of a slot-machine based game. Alternatively, the method could be implemented using a stand-alone computer, in which a standalone application would do the game processing of the card data and display the output in graphical form to the user. While an online (or Internet-based) embodiment has been described most particularly, it will be understood that the system could equally well be implemented for a single-user on a stand-alone computer.

[0060] The foregoing is considered as illustrative only of the principles of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact processes, components and applications shown and described, and accordingly, all suitable modifications and equivalents may be resorted to, falling within the scope of the invention and the appended claims and their equivalents.

What is claimed is:

1. A method for determining a winning card in a multi-pattern computerized bingo game, comprising:

- a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;
- b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;
- c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;
- d. calling the call numbers in the sequence to construct, for each card, a daub matrix of grid positions corresponding to any card numbers which match the called number, and, for each called number:
  - (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and

- (ii) determining from each product matrix whether the card matches the pattern; and

- e. repeating steps (i) and (ii) at least until such time as a pattern match is determined for any pattern, the card being thereby determined as a winning card, and determining how many called numbers must be called to generate said winning card.

2. The method of claim 1, wherein steps (i) and (ii) are repeated in a game until a predetermined number of winning cards is reached.

3. The method of claim 1, wherein steps (i) and (ii) are repeated in a game until a predetermined number of called numbers is called in the sequence, regardless of the number of winning cards determined.

4. The method of claim 1, wherein each daub matrix comprises a matrix of 0s and 1s, each 1 representing a match in the grid position.

5. The method of claim 4, wherein the determining step comprises evaluating each grid position in the product matrix and, finding a pattern match if values stored in the grid positions making up the pattern are greater than zero.

6. The method of claim 1, wherein only one winning card is determined per game.

7. The method of claim 1, wherein the method continues after one winning card is determined.

8. The method of claim 2, wherein the number of winning cards is predetermined by an operator.

9. The method of claim 3, wherein the number of called numbers is predetermined by an operator.

10. The method of claim 1, wherein the sequence comprises 75 call numbers.

11. The method of claim 1, wherein the sequence comprises 90 call numbers.

12. The method of claim 1, wherein the bingo game is accessible to players over a distributed network, a game server being programmed to determine the winning card or cards for communication to a plurality of client computers, and each client computer being programmed for displaying a game re-creation and announcing the winning card or cards.

13. The method of claim 12, wherein the game re-creation is interactive.

14. The method of claim 1, wherein the multiple patterns are set by an operator.

15. The method of claim 1, wherein the multiple patterns are set by a player.

16. The method of claim 12, wherein the multiple patterns are displayed graphically during the game re-creation on the client computers.

17. The method of claim 12, wherein the game re-creation includes displaying, for each card, a "number away" indication.

18. The method of claim 13, wherein the game re-creation provides an auto-daub feature.

19. The method of claim 13, wherein the game re-creation provides a manual daub feature.

20. The method of claim 19, wherein the game re-creation provides a catch-up feature.

21. A method for determining whether any winning cards have been selected in a multi-pattern computerized bingo game, comprising:

- a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;
- b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;
- c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;
- d. calling the call numbers in the sequence to construct, for each selected card, a daub matrix of grid positions corresponding to any card numbers which match the called number, and, for each called number:
  - (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and
  - (ii) determining from each product matrix whether the card matches the pattern, the card being thereby determined to be a winning card; and
- e. repeating step (d) until a predetermined number of call numbers have been called, and determining how many called numbers must be called to generate a winning card, if any winning cards can be determined in the predetermined number of call numbers.

22. A method for determining a winning card in a multi-pattern online bingo game accessible to players over a distributed network, comprising programmed steps for execution on a server, the steps including:

- a. setting multiple patterns to be played for simultaneously in the bingo game, each said pattern being stored in a computer memory as a pattern matrix;
- b. generating call numbers from a range of available numbers, and a sequence in which the call numbers are to be called, the call numbers and the sequence being stored in the computer memory;
- c. providing at least one bingo card for selection by a player, the card comprising card numbers from the range, the card numbers being associated with virtual positions on a grid, and, storing the card numbers and the grid positions thereof in the computer memory;
- d. calling the call numbers in the sequence to construct, for each card, a daub matrix of grid positions corresponding to any card numbers which match the called number, and, for each called number:
  - (i) multiplying the daub matrix by each pattern matrix to produce a product matrix for each pattern; and
  - (ii) determining from each product matrix whether the card matches the pattern; and
- e. repeating steps (i) and (ii) at least until such time as a pattern match is determined for any pattern, the card being thereby determined as a winning card, and determining how many called numbers must be called to generate said winning card, wherein said server is programmed for communicating the winning card or cards to a plurality of client computers, each said client computer being programmed for displaying a game re-creation to announce the winning card or cards.

\* \* \* \* \*