(54) Title: ELECTRONIC BOARD GAME HAVING CLUE DELIVERY CIRCUITRY

(57) Abstract

This invention is a universal game housing (10) having switch input means (12) comprising a matrix of touch sensitive input switches; interpretation circuit means (15) connected to said matrix of input switches for receiving and interpreting user switch input, and for generating a game input based thereon; and, game progression circuitry for receiving the game input from said interpretation circuit means (15), and for determining game progress based thereon. The housing (10) is additionally equipped with PLAY switch means (14) for activating the game, power source means (13) connected to the PLAY switch means (14), and speaker means (18) for generating audible game play responses. Several specific embodiments of the invention include the game progression circuitry adapted to perform game specific calculations, generate game play responses based upon the calculations, and the method of implementing specific game play embodiments.
### FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>KE</td>
<td>Kenya</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>CS</td>
<td>Czechoslovakia</td>
<td>LV</td>
<td>Latvia</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TJ</td>
<td>Tajikistan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UG</td>
<td>Uganda</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
</tbody>
</table>
ELECTRONIC BOARD GAME HAVING CLUE DELIVERY CIRCUITRY

Field Of The Invention

The invention relates to the field of electronic board games. More specifically, it relates to an electronic game board, comprising a patterned game board overlaying a matrix of input switches and the associated circuitry for implementing random and cued game play, and the method of game implementation utilizing same.

Background Of The Invention

The electronic game industry has developed along several parallel pathways which mirror the development of high speed, high density semiconductor circuit development and related computer development. On-board processors perform multiple tasks including receiving and interpreting user input, calculating responses to user input, and generating audible and/or visual display of the game play.

Hand-held semiconductor driven board games provide entertainment and, nominally eye-hand skill development for older, more dexterous players; while basic push-button games entertain younger players having less sophisticated motor skills. Each game has required development of a unique board, unique software for providing the game progress, and a unique means for receiving and interpreting player input.

Ideally, a toy designer and/or manufacturer would like to develop a universal input device having uniform means for reading and interpreting input to the device, which device could be adapted, via individualized software and associated graphic overlays, to any number of a plurality of game plays.

It is therefore an objective of the present invention to provide a universal game housing having a plurality of surface mounted input switches in a matrix and having universal circuitry for receiving and
interpreting input to the game, said input being provided via contacting one or more of the input switches through an over laying membrane having a game pattern printed thereon.

It is a further objective of the invention to provide the above-described game board further equipped with customized means for performing game-specific calculations, based in part on switch input to said matrix, and for generating responses and game play based on said calculations.

Yet another objective of the invention is to provide a game play process flow which returns to the starting point, thereby eliminating the need for a marker.

Summary Of The Invention
These and other objectives of the invention are realized in a universal game housing having switch input means comprising a matrix of touch-sensitive input switches; interpretation circuit means connected to said matrix of input switches for receiving and interpreting user switch input and for generating a game input based thereon; and, game progression circuitry for receiving the game input from said interpretation circuit means and for determining game progress based thereon. The housing is additionally equipped with PLAY switch means for activating the game, power source means connected to the PLAY switch means, and speaker means for generating audible game play responses. Several specific embodiments of the invention are detailed wherein the game progression circuitry is adapted to perform game-specific calculations and generate game play responses based upon said calculations. The method of implementing three specific game play embodiments of the invention is included herein.

Brief Description Of The Invention
The invention will now be described in greater detail, with specific reference to the drawings wherein:
Figure 1-1 provides a top view of the inventive universal housing including a matrix of input switches, speaker means and ON/OFF switch means.

Figure 1-2 provides a side view of the universal housing of the present invention.

Figure 1-3 provides a rear view of the universal housing including power source and microprocessor locations.

Figure 2 is a schematic circuit diagram of the invention including a matrix of input switches, ON/OFF switch means, speaker means, related circuitry, and microprocessor means.

Figure 3 provides a basic process flow utilized by all game play embodiments of the invention.

Figure 4 illustrates a game housing overlay for a first game play embodiment of the invention.

Figure 5 illustrates a representative look-up table for use with the first game play embodiment of the invention.

Figures 6-1 through 6-4 provide a flow chart depicting a representative process flow for the first game play embodiment of the invention.

Figure 7 illustrates a game housing overlay for a second game play embodiment of the invention.

Figures 8-1 through 8-11 provide a flow chart depicting a representative process flow for the second game play embodiment of the invention.

Figure 9 illustrates a game housing overlay for a third game play embodiment of the invention.

Figures 10-1 and 10-2 provide a flow chart depicting a representative process flow for the third game play embodiment of the invention.

**Detailed Description Of The Preferred Embodiment**

Several terms which will be used throughout the following description are set forth in the following glossary:
microprocessor: a microprocessor, such as the T150C19 from Texas Instruments, programmed utilizing customized masks to include customized game progression circuitry, and further including input interpretation circuitry, timing means, Read Only Memory (ROM) storage, and Random Access Memory (RAM) storage.

input switch: one of the array of input switches, using pressure-sensitive contact switches;

input switch array: an array of input switches, representatively illustrated as a 6 x 6 matrix of input switches;

switch input: entry of user input to one of the input switch array by depression of one of the input switches;

input interpretation circuitry: circuitry and related software for determining from which of the input switches the switch input has been received, for judging if the switch input is valid, and for generating a game input based upon valid switch input;

game input: input from the input interpretation circuitry to the game progression circuitry upon judgment that the received switch input is valid; and

game progression circuitry: circuitry for receiving game input, for determining game play progression, and for generating outputs, including audible responses, based upon same.

As diagramatically illustrated in Figures 1-1 through 1-3, universal game housing 10 includes PLAY switch 14 for manual activation of the game. The PLAY switch means 14 is connected between the power source 13, ideally a battery source, and the microprocessor 15. Figure 1-2 and Figure 1-3, which offer side and rear views of housing 10, illustrate a enclosure for power source 13 which is connected to the PLAY switch means 14. A short time (less than 25ms) after the battery supply has been connected to the circuit, the
voltage on the INIT line will stabilize at the positive battery supply voltage. In this state, the capacitor C1 is charged to near positive battery supply voltage. When the 'Play' Switch is released, the voltage on the INIT line and the capacitor C1 begins to increase towards the positive battery Supply. The length of time it takes for the capacitor C1 voltage (and INIT line voltage) to reach positive battery supply voltage is dependent on the value of resistor R1 and capacitor C1.

The IC (UI) will "wake up" and begin executing software when the voltage on capacitor C1 (and the INIT line) reaches a threshold value near the positive supply voltage (less than 25ms). Upon initialization by the microprocessor, game progression circuitry therein selects a target for game play and prompts the user to begin play. In addition, the input interpretation circuitry of the microprocessor begins polling pins PI-P10, which are selectively connected to the input switches 12, to ascertain if input from a row and a column complete a circuit indicative of switch input.

The universal game housing 10 has an input switch array comprising a matrix of pressure-sensitive input switches 12, representatively provided in a 6 x 6 switch matrix.

Although a differently shaped array and/or an array having a different number of input switches may be implemented, the three specific game play embodiments detailed hereinafter have been adapted for play on the 6 x 6 matrix input switch array, and all related description will incorporate such an array.

One skilled in the art could readily modify the housing's input switch matrix and customized game play to utilize an alternative input switch array without departing from the teachings of the present invention.

The input switches are ideally membrane pressure or touch sensitive switches for use with the later-described game play. A membrane switch, in
accordance with known technology, comprises a membrane carrying a pattern of moveable contacts overlaying a corresponding pattern of fixed contacts. One set of contacts is interleaved and connected to an associated circuit means. The other set of contacts is constructed so that when pressure is applied to the membrane, the two sets of contacts touch each other completing the circuit between the interleaved contacts. Capacitive switches, discrete microswitches, direct PCB inputs through a silkscreened array of input locations, or other low profile switching implementation may alternatively be incorporated for use in the universal housing. Adaptation of the input circuitry, for translating input to the alternative switches and switch array into coordinate values paralleling the matrix locations of the preferred embodiment, would be evident to one having skill in the art. For pressure or touch sensitive switches, the physical entity which depresses the input switch may be the player's finger, a game piece, or other apparatus having dimensions appropriate to contact only one of the input switches at a time. Should capacitive switch means or other electrically activated switch means be incorporated into the game housing, however, the use of specific switch depression means may be more limited.

As illustrated in the circuit diagram of Figure 2, each switch 12 is located above the intersection of two leads and thence to the microprocessor 15. Upon depression of the membrane covering one input switch 12 of the array, the two leads come into contact with each other and the input is provided to two or more input pins of the microprocessor, in a manner by which a determination can be made as to which of the input switches has been depressed by the user. Leads connected to pins P1-P6 provide input from each of the rows in the matrix array. Input from four of the columns are provided directly to pins P7 through P10,
while input from the remaining two columns are provided through diode pair D1 and D2 or diode pair D3 and D4 to pin pair P10 and P9 or pin pair P8 and P7, respectively.

In this manner, in spite of the fact that the microprocessor has fewer than 12 input pins, distinguishing inputs can be provided for discrimination by the microprocessor. For example, while input to an input switch located in the third column from the left will put activity on pin P10, input to an input switch in the first column from the left will put activity on both pins P10 and P9. Resistors R2-R5 are "pull up resistors, pulling the pins to the positive battery supply voltage. Depression of a switch, and the attendant sensing at two or more pins will be of a low value on the read line.

Input to one of the input switches in the input switch array is "interpreted" by the input interpretation circuitry to isolate and identify the individual input switch which has been selected by the user. Thereafter, the input interpretation circuitry performs input validation to determine if the switch input is valid, first by applying a debounce routine in accordance with state-of-the-art processing. Further input validation may be implemented as appropriate for the ensuing game play, including a determination as to whether the switch input is different from the previous switch input, in order to avoid interpretation of prolonged switch closure as a series of switch inputs. Another input validation analysis may invalidate switch input received during another player's turn or while a non-interruptable program step is executing.

Upon determination that switch input is valid, the output of the input interpretation circuitry (now designated as game input) is provided to the game progression circuitry of the microprocessor. The game progression circuitry generates a game play response based upon the received game input and the relevant game
play programming. In so doing, the game progression circuitry may refer to one or more look-up table locations for selecting an appropriate response to the user's input or may randomly access one of its stored series of responses. Alternatively, the game progression circuitry may utilize the input in a calculation for ascertaining an appropriate response to the input. For the preferred embodiment, utilizing the TI50C19, the amount of available ROM storage is 32 kbytes. The ROM provides storage for response look-up tables, sound storage, and fixed programming routines. In addition, 128 bytes of RAM is available for temporary storage of game play inputs (e.g., target location, registers and counters) as required by the specific game play embodiment.

After a response has been determined, the game progression circuitry (also referred to as "the system" hereinafter) accesses the appropriate sound from its storage location in the associated RON and provides it to audio outputs A1 and A2 for audible reproduction via speaker means 18. The output of the game progression circuitry (i.e., the selected response) may additionally be provided to the RAN for temporary storage as required by the game play, as will be further detailed below.

In addition to the input interpretation circuitry, game progression circuitry, ROM, and RAM, the microprocessor has a timing function for deactivating the game if a preset time period has elapsed between valid switch inputs. The timing function is executed by resetting a timer after any switch input which has been judged to be valid. If the timer has not been reset, and the preset time period runs, a software instruction is generated to deactivate the game (i.e., stop polling pins PI-P1O for inputs and stop executing the program) until the PLAY switch is again depressed.

A basic operating routine for the game progression circuitry for all game play embodiments is
illustrated in Figure 3. Upon player closure of the PLAY switch, and the consequent sensing of input at the INIT pin of the microprocessor, the system performs its Initialization function at box 101. For all embodiments, initialization includes the steps of starting the timer, beginning polling at pins P1-P10, and choosing a target in accordance with the game play programming. The target is a numerical value which corresponds to a location on the gameboard and is chosen randomly by the microprocessor at the beginning of each game. The object of all of the specific embodiments is to locate the target, either through a series of switch inputs to successive switch locations leading to the target or by isolated switch inputs based upon random selection or responses to clues.

The next step is input interpretation involving the two decision boxes 102 and 103. At box 102, the system senses switch input. If a switch input is detected (i.e., a "Yes" response), the system then determines if the switch input is valid, at decision box 103. If the switch input is judged to be valid (i.e., another "Yes" response), the next step, at 104 is to execute the game play process flow. If the output of either decision box 102 or 103 is a "No" determination, the timing function is invoked. At decision box 105, the determination is made as to whether the preset time period has expired. If the time period has expired, the system performs its Shut Down function, at box 106.

The details of the initialization step, including target selection, the game play process flow execution, and the shut down step will now be detailed for each specific game play embodiment.

FIRST EMBODIMENT—THE GUESS GAME

The "GUESS" game provides a series of randomly accessed clues to the identity of the selected target. As illustrated in Figure 4, the game overlay comprises 35 pictures and a HELP graphic (the question mark) over
the 36 switch 10 locations in the matrix. Each of the pictured items has attributes relating to its size, loudness, hardness, color and speed. A numerical value is preassigned to each of the items for each of the attributes. The table illustrated in Figure 5 provides a representative look-up table of numerical attribute 15 values for the pictured items. For example, items such as a jet plane or a ship have high numerical values for size (12 and 13, respectively). The jet plane also has a high numerical value (7) for speed, while the ship's speed value is considerable lower (4). If an attribute is not applicable to 20 the item (e.g., a crayon cannot have speed and a chair cannot make noise) or if the attribute value is low for that item a snowflake isn't very hard), then the assigned numerical value will be zero. Color values are based upon a listing of 6 available colors, with numerical values from 1-6.

If the item is multicolored (such as a butterfly) or otherwise difficult to characterize (e.g., the color or a lion may be considered to be brown or yellow), then the item will be assigned a numerical color value of zero. Additional features are stored for particular items, as evidenced by the HELP value listed for some of the pictured items. Representatively, a numerical HELP value of 1 may mean that the item is good to eat, 2 may mean that the item has eyes, 3 may mean that the item has wings, 4 may mean that the item has wheels, 5 may mean that the item has wings and eyes (2+3), and 0 may be assigned to all other items for which those attributes do not apply.

The numerical attribute look-up tables are stored in the ROM of the microprocessor. In addition to storage of the table of numerical attribute values for each pictured item, the ROM also stores a plurality of bit sequences representing the various audible clues or responses which can be retrieved and output at speaker 18.
A first category of stored sounds are bit sequences for prompting user input, such as "WHAT AM I?" or "C'MON, TAKE A GUESS." Another category of stored sounds includes the bit sequences for generating audible clues based on comparisons between the numerical attribute values of the target item and the player selected item. Such audible clues include the following: "I'M -- BIGGER, SMALLER, ABOUT THE SAME SIZE AS, LOUDER, QUIETER, FASTER, SLOWER, HARDER, SOFTER-- THAN THAT" or "I'M -- A DIFFERENT, THE SAME-- COLOR AS THAT" or "THAT'S -- WHITE, RED, YELLOW, GREEN, BLUE, GREY--, I'M NOT." Additional audible clues may be generated based upon the HELP values of the respective items. Examples of HELP clues include "I'M GOOD TO EAT" and "I HAVE -- EYES, WINGS, WHEELS--, THAT DOESN'T." The HELP clues are generated in response to switch input at the HELP switch (with the question mark overlay) when the player wishes to have information in addition to the responses generated based upon switch input and the resulting numerical attribute comparison. Finally, there is a category of sounds generated subsequent to game play, including such phrases as "YOU'RE RIGHT!" and "YOU TOOK ONLY THREE TRIES!"

Multiple sound bit sequences are accessed for inserting different phrases or sentences to compile a response. As is evident from the foregoing examples, sounds such as "I'M" and "THAN THAT" are used repeatedly in response to different switch inputs. Therefore, it is more efficient, from a storage space perspective, to store only one bit sequence representing the word or phrase which is used repeatedly and to access it in conjunction with other bit sequences in order to formulate an appropriate response.

With reference to the process flow detailed in Figure 6, the GUESS game initialization, 201, includes clearing RAM table locations of any previously stored variables, starting the timer, beginning polling of pins
P1-P10, and playing introductory audible effects. Thereafter, the next initialization step is random choice of a target and storing the switch location coordinates of the chosen target in the RAM. Once the target has been chosen, the numerical attribute values for the target item are retrieved from the ROM and the variables are provided for temporary storage in the cleared table location of the RAM. Each stored target variable is preceded with a "T" for "target" throughout the process flow chart for ease of description. Initialization for the GUESS game is now complete and the system waits for either player switch input or expiration of the preset time period.

Upon detection of switch input and determination that the switch input is valid, at input interpretation box 202, the game progression circuitry for the GUESS game evaluates the input, which is designated as "N" in the process flow chart. The N value of a switch is based upon its X and Y coordinates in the matrix, from X=1, Y=1 at the switch designated as N=1 to X=6, Y=6 at the switch designated as N=36. The N value for each X and Y coordinate pair can either be stored in a look-up table or calculated by applying the function X+6Y-6=N. When the game input of the selected item N has been received and communicated to the game progression circuitry, the circuitry first makes a comparison of the X-Y coordinates of the switch location for item N to those stored for target item T to determine if the selected item is the target (N=TARG?), at decision box 203. If N is the target, the game progression circuitry accesses the bit sequence for the "WIN" theme and outputs same at speaker 18, at box 204. If the target has been located during the player's first turn, as determined at decision box 205, the game progression circuitry also plays a "LUCKY GUESS" message at 206. If the player took more than one turn, the incrementing of which is discussed further herein, the
game progression circuitry will issue a different audible message based upon the number of turns taken as seen in boxes 207 through 214.

After the turn has been completed and appropriate responses have been generated, the game progression circuitry accesses a prompting message, at 215, inviting the player to play again. At this juncture, the player must again press the PLAY SWITCH TO re-initialize the game, causing resetting of the timer and choice of another target. As in the Figure 3 process flow, if no switch input is received, or if received switch input is not judged to be valid input, during the preset time period, the system will perform its shut down function. Generally, the shut down function simply involves deactivating the software to stop polling the pins, to disable the time, and to "go to sleep" until the PLAY switch is gain depressed. In addition, the microprocessor may generate a prompting message after expiration of a first time period and then perform the actual shut down procedures upon expiration of a second, longer time period.

When the output of decision box 203 (N=TARG ?) is "No," the game progression circuitry evaluates the game input of the players' selection of item N. After the determination has been made, at decision box 203, that the selected N is not the target, the decision is next made, at box 250, as to whether N is 36 (i.e., switch input has been made to the HELP button). If the switch input has been to the HELP button, the play progresses along the process flow illustrated in Figure 6-4.

At box 251 of Figure 6-4, a randomly chosen value from 1 to 4 is placed in the R-register. Next the system determines if the numerical color value for the target is zero, at box 252. If the color value is not zero, then the HELP value is evaluated at 253. If the HELP value is zero, the system generates a color message
at boxes 270 and 271, by accessing a color message for
the color of the target (e.g., "I'M BLUE"). When the
HELP value is not zero, the system next checks the value
in the R-register at 254. If the R value is 4, the
color message is generated, again at boxes 270 and 271.

When the R value is not equal to 4 as
determined at decision box 254, or when the target color
value is zero as determined at box 252, the system next
checks the numerical HELP value. The message generated
depends upon the numerical HELP value, 256 through 260,
for the selected target. As correlated to the
representative HELP values recited above, when the HELP
value is 1 as determined at 255, the message generated
at 262 and 263 may be "I'M GOOD TO EAT." For the HELP
value of 2, as well as the HELP value of 5, when the R
value is less than 2 as determined at box 261, the
message generated at 264 and 265 may be "I HAVE TWO
EYES." For a HELP value of 3, as well as the HELP value
of 5, when the R value is greater than 2 as determined
at box 261, the message generated at 266 and 267 may be
"I HAVE WINGS." Finally, when the HELP value is 4, the
message generated at 268 and 269 may be "I HAVE
WHEELS." After the appropriate message has been
accessed and output at speaker means 18, the system
returns to await input, at box 202 of Figure 6-1.

If the determination has been made that N is
not 36, the game progress circuitry follows the process
flow depicted from box 216 of Figure 6-1 through
Figures 6-2 and 6-3. While the process flow indicates a
perceived order for generating attribute clues, in fact
the software accesses the various attributes and
performs comparisons based thereon largely in a random
order, subject only to a few exceptions which will be
discussed further below. The variables for selected
item N are retrieved from their ROM storage location and
placed in a RAM table and a random value R is placed in
a RAM register (hereinafter referred to as the
R-register), at box 216. The comparisons will be made based upon the random value of R, which is frequently updated.

At decision box 217, the system checks whether the random value of R in the R-register is 1. If R does not equal 1, the system progresses down to determine if R equals 2, 3 or 4 at boxes 223, 224 and 225, respectively. If R is 1, the system multiplies the numerical color value of the selected item by the numerical color value of the target item and determines if the result is greater than zero, at 218. When the result is greater than zero, the system next compares the numerical color value for the selected item to the numerical color value of the target item, at 219.

Should the color values be the same, the game progression circuitry puts a value of 2 in the clue register, at box 220 and generates a color clue in the form of an audible message (e.g., "I'M THE SAME COLOR") at box 244. If, however, the color values differ, the game progression circuitry puts a value of 1 in the clue register, at box 221, and generates a different color message (e.g., "I'M A DIFFERENT COLOR") at box 244.

If the color value of either the target item or the selected item is zero, and therefore the result at decision box 218 is "Yes" (i.e., the result of multiplying COLR X TCOLR = 0), the system adds a random integer (2, 3, 4 or 5) to register R, at box 222, and the process flow continues.

When the random value R in the R-register equals 2, as determined at box 223, the game progression circuitry next compares the numerical size values for the items, at box 226. If the size values are equal, the game progression circuitry generates a same size message ("I'M ABOUT THE SAME SIZE") and puts a 3 into the clue register at 227. Should the size values be unequal, a determination is made at decision box 228 of whether the numerical size value of the selected item is
larger than the numerical size value of the target item. If the result of the decision indicates that the selected item is larger than the target item, the game progression circuitry puts a 4 into the clue register, at box 229 and generates a smaller message ("I'M SMALLER THAN THAT") at 244. Should the target item be larger than the selected item, the game progression circuitry puts a 5 into the clue register, at box 230 and generates a larger message ("I'M LARGER THAN THAT") at 244.

When the random value in R-register is 3, as determined at decision box 224, the game progression circuitry analyzes the loudness attributes of the items by determining if the numerical loudness value of the selected item is greater than the loudness value of the target at decision box 231. If the selected item has a larger loudness value as that of the target item, the system, at box 232, adds a 6 to the clue register and generates a quietness message ("I'M QUIETER THAN THAT") at 244. When the loudness value of the target item exceeds that of the selected item, as determined at box 233, then the system puts a 7 into the clue register at box 234 and generates a loudness message ("I'M LOUDER THAN THAT") at 244. Should the loudness values be equal, such that a "No" decision results from decision boxes 231 and 233, the system puts an R value of 2 into the R-register, at box 239, and the system defaults to the size analysis detailed above with reference to boxes 226-230. Since the size analysis can compare any two objects, whether bigger, smaller or about the same size, it is accessed whenever the other attribute comparisons cannot result in generation of a clue.

When the random R value in the R-register is 4, as determined at box 225, the game progression circuitry performs a hardness analysis. If the numerical hardness value of the selected item is greater than the numerical hardness value for the target item, as judged at
decision box 235, the system puts an 8 into the clue register at box 236 and generates a softness message ("I'M SOFTER THAN THAT") at 244. Should the numerical hardness value of the target exceed that of the selected item, as judged at decision box 237, the system puts a 9 into the clue register at box 238 and generates a hardness message ("I'M HARDER THAN THAT") at 244. As with the loudness determination, the system defaults to the size analysis by putting an R value of 2 into the R-register, at 239, when the hardness values of the two items are the same (i.e., a "No" output from both decision boxes 235 and 237.

If the R value is none of 1 through 4, the game progression circuitry does a speed analysis, comparing the numerical speed values of the selected and target items. If the determination is made at decision box 240 that the speed value for the selected item exceeds that of the target item, the system places a 10 into the clue register at box 241 and generates a speed message ("I'M SLOWER THAN THAT") at 244. On the other hand, if the numerical speed value of the target item exceeds that of the selected item, as judged at decision box 242, the system will place an 11 in the clue register at box 243 and generate a different speed message ("I'M FASTER THAN THAT") at 244. As above, if the results of the decisions at boxes 240 and 242 indicate that the numerical speed values are the same, the system defaults to the size determination by putting a 2 into the R-register at 239.

When generating a message at 244, the game progression circuitry adds 10 to the clue register and accesses the message segment "I'M" and the appropriate clue. If the value in the clue register is greater than 14, as determined at 245, a different type of message is generated (e.g., "THAT MOVES, I DON'T") at box 246. After the message has been generated, the "fill" message (e.g., a musical theme played between successive inputs)
is generated and the system returns to await input at
202.

As discussed above, additionally available to
the player is the option of obtaining an extra clue by
contacting the HELP switch at any time during the game
play, with the exception of those times when the system
determines that switch input is not valid, for example
due to the simultaneous execution of a non-interruptable
message. Input to the HELP key causes the system to
access the HELP value which has been stored for the
target item and to generate the appropriate audible clue
based upon same.

SECOND EMBODIMENT—THE TAXI GAME

The TAXI game, the overlay for which is
representatively provided in Figure 7, has two taxi
icons, associated with taxi telephones, along with a
plurality of destinations, comprising 5 houses and 6
stores, including 4 shops, an airport and a train
station, each located over a switch. Another switch
icon is a gas station, switch input to which will be
discussed hereinafter. Players are prompted to bring
their taxi, either by moving their finger or by moving
an optional game piece, along the roadway from their
taxi icon to the prompted destination. The game
progression circuitry judges the movement of the
player's taxi by evaluating the switch closures as each
switch along the chosen pathway is depressed. The game
progression circuitry stores the X-Y coordinates of each
move along with the N value of the contacted switch
(from N=1 under the taxi telephone located in the upper
left corner of the matrix, where X=1 and Y=1, through
N=36 under the taxi telephone located at the lower right
corner of the matrix, where X=6 and Y=6) for use in
determining if the move or series of moves is
permissible. RAM registers store values related to
whether the player is taking his first or a successive
turn (LIV register), the N switch locations of the
destinations including the stores (which includes the shops, the train station and the airport) and the houses (the Designation register), the amount of gas that the player/taxi has and the resulting audible output (GAS and PUTT registers), the location of the last switch closure (LASTX, LASTY, and, LASTN), the location of the randomly selected target (the assigned value 1-5 for the house target, hereinafter referred to as the House register, and the assigned value 11-16 for the store target, hereinafter referred to as the STORE register), a KILL register for negating switch input to all but the phone switches until each new turn is initiated, a STEP register for tracking whether it is the first or second step of a player’s two-step turn, and a FRESH register, for tracking arrival vs. departure at a destination, should a player provide input to a single location two times in his two-step turn. Where applicable, there are two of each kind of register in order to store the variables separately for player 1 and player 2.

For Initialization of the TAXI game, as shown in Figure 8 at box 301, the microprocessor conducts the following procedures: putting a zero into a RAM registers which will be used for storing variables throughout the game play; accessing the bit sequence or sequences for producing the audible game theme at speaker 18; starting the time; beginning polling of the pins P1-P10; and, putting a value of 1 in the KILL register. The audible game theme provides the sound of a telephone ringing and the input interpretation circuitry awaits switch input at one of the two telephones (at N=1 or N=36). Switch input to any other switch is judged to be invalid at this juncture of game play.

Upon receipt of switch input from switch 1 or 36 at box 302, the input interpretation circuitry determines which of the taxi telephones has been contacted, at box 303. One of the taxi telephones is
designated as player 1 while the other is designated as
player 2, for the purposes of tracing turns and
accumulating certain variables which will be further
discussed herein.

Regardless of which player (i.e., taxi
telephone switch) has been chosen, the ensuing "start
up" process flow is identical. First the system
accesses the value in the LIV register for the
respective player, at boxes 304 and 308. If the LIV
register has a zero value, indicating that it is the
player's first turn, the game progression circuitry
checks the value of the player's GAS and PUTT
registers. If the PUTT register indicates, at decision
boxes 305 and 309, that the player has low fuel, fuel is
added. A value of 5 is placed in the PUTT register at
the beginning of a turn, as indicated at boxes 306 and
310, and the attendant audible message will be a fast
"putt-putt" sound. The GAS variable is a "stepper,"
used to cause the engine sound ("putt-putt") to appear
to slow down. The GAS variable is set at zero at the
start of any turn. Lastly, the system puts the
"identity" of the player into the LASTX, LASTY and LASTN
registers, at boxes 307 and 311. If player 1 has been
selected, switch N=1 at X=1 and Y=1 is the appropriate
designation. In addition, a value of 1 will be placed
in the 1LIV register, at 307. If player 2 had been
selected, switch 36 at X=6 and Y=6 is entered at 310,
along with a value of 1 being placed in the 2LIV
register.

At this juncture, the target is selected at box
312. Randomly, one of the house values, 1-5, is placed
in the House register, while one of the store values,
11-16, is placed in the Destination register. As
depicted in the representative process flow, for each
successful player turn, there are two steps involving
one house destination and one store destination with the
first step of a turn being to the house destination.
Clearly, the destinations can be randomly selected from any of the house or store destinations and the two-step selection process can be modified for random choice of any destination other than the current location of the player's "taxi."

Once a first target is selected, a zero value is placed in the SUMDIF, KILL and STEP registers and a 1 is placed in the FRESH register. At box 313, the system accesses the bit sequences to produce an audible message instructing the player to pick them up at the randomly selected destination (see: "PLAY '1ST GO (1, 2, or 3)"), which for the representative game play process flow indicates that one of three stored messages can be generated to instruct the player to pick someone up, and that message is concatenated with either a ()HOUSE message which identifies the house destination by the number printed on the graphic overlay or a (_+5)HOUSE message which identifies the house destination by the color of the house on the graphic overlay). Thereafter, the system plays the appropriate PUTT message and awaits player input of their first move.

If no new input has been detected, at decision box 314 (as in the process flow for all embodiments at Figure 3), the system determines if the preset time period has expired. When the time period has expired, the system plays a prompting BEEP message at 317 and thereafter shuts down. As discussed above, there may be two time periods included in the shut down routine, a first T1 after which the audible message is generated, and a second T2 after which the system shuts down.

Whenever new input is detected at decision box 314, the system determines if the switch input has been received from the taxi at switch N=1, where X=1 and Y=1, by judging if X+Y=2, at box 316. If the decision is
that X+Y is not equal to 2, then the system checks to see if X+Y=12 to determine if switch input has been received for the taxi at switch N=36, where X=6 and Y=6. The system thereafter verifies if it is the turn of player 1 or player 2. The player must now move by providing input to a new switch. Input to the telephone icon switches N=1 or N=36 at this juncture will be ignored if the input is to the wrong player's phone or will result in a repeat of the destination message if the same phone icon is again contacted.

If the switch input has been received at neither N=1 or N=36, the input will be further evaluated to determine the player's location and the permissibility of that location. The system checks the KILL register at 322. Should the value in the KILL register be a 1, the input is disregarded and the system awaits new input at 314. If, however, the answer at decision box 322 is "No" i.e., KILL≠1), the game progression circuitry loads the player's current X and Y coordinates for application in the calculation (X+6Y-6), at 323.

The result of the calculation (X+6Y-6) provides an N switch value representing the current location of the player. The N value is compared at decision box 324 to the locations stored in the aforementioned table of stores and houses. If the calculated N value is not found in the table, the system continues with the MOVER process flow detailed below with reference to Figures 8-7 through 8-10, under the assumption that the player is still en route to a destination. On the other hand, if the calculated N is found in the table, the system first determines whether the value of the FRESH register is a 1, at decision box 325. If the value of the FRESH register is 1, the system stores the player's current X coordinate in the LASTX register, the current Y coordinate in the LASTY register, and the calculated N value in the LASTN register, at 326, and then returns to
box 314 to await new input. However, if the value in
the FRESH register is not equal to 1, the system
continues with the STOR process flow detailed
hereinafter with reference to Figures 8-3 through 8-6.

The MOVER process flow, depicted from the
Figure 8-7 through Figure 8-10, begins by checking, at
box 350, if N=5, the switch location of the gas
station. A move to N=5 is a permissible move from an
adjacent switch location unless the LASTN value is 11
(i.e., the "taxi" moved from house 2 at switch 11 to the
gas station without using the roadway; but presumably
through "the wall" of house 2). If the LASTN value was
11, the game progression circuitry stops the play and
plays the CRASH message at 352, determines if player 1
or 2 initiated the illegal CRASH move at 353, and puts a
zero into the appropriate LIV register, 1LIV at box 354
or 2LIV at box 355. The turn ends as the system resets
by putting a zero into each of the STEP and WHO
registers and a 1 into the KILL register at 356. The
system awaits new input as at box 301.

Decision boxes 357 through 367 also determine
if moves are permissible, or if the taxis are making
"CRASH" moves "through walls", based upon a comparison
of the current switch input location N to the previous
switch input location LASTN. Accordingly, if, for
example, it is determined that the taxi moved from the
roadway at switch 3 to the train station at switch 2
(Box 357) without the intermediate moves along the
roadway (i.e., from switch 3 to switch 9, to switch 8,
and then to switch 2), the play stops and the CRASH
message is played at 353, the appropriate player's LIV
register is given a zero value at 354 or 355, and the
turn is terminated at 356.

The remaining boxes 358-367, for which a "Yes"
decision leads to execution of the steps found in boxes
352-356, represent a series of impermissible CRASH moves
which are based on player switch input to a target
switch from an adjacent roadway switch location via an improper direction.

For each of the decision boxes 350 and 357-367, a "No" decision, representing a determination that the move is not an impermissible crash move, results in the game progression circuitry continuing to evaluate the player switch input as the player moves. At decision box 368, the system checks value in the FRESH register. If the value in the FRESH register is a 1, the system, at box 369, stores the current X coordinate in LASTX, the current Y coordinate in LASTY, the current N in LASTN, and changes the value in the FRESH register to zero. The system checks, at box 383, to see who made the move and then plays the appropriate PUTT sound for the player, at 384 or 385, while the game continues and the system awaits the player's next input, at 314.

If the FRESH value is not 1, the game progression circuitry executes a sum:difference calculation to determine the difference between the last X and Y coordinates for the player (LASTX and LASTY) and the present X and Y coordinates, at 370. If the result of the calculation (referred to as SUMDIF) is zero, as determined at 371, such is an indication that the player has depressed the same switch as had previously been evaluated by the game progression circuitry. Under those circumstances, the system will again determine the identity of the player and access the appropriate PUTT message for that player while awaiting new input, as discussed with reference to boxes 383-385 and 314, above.

If the SUMDIF is greater than 3, as determined at decision box 372, the system stops play and generates a SIREN message at 373, then follows the process flow at the end of a player's turn, as described above with reference to boxes 353 through 356. There is an assumption that, particularly with young players, while moving a playing piece or their finger along the
gameboard, the players may not physically press down on the membrane enough to contact each Switch. Therefore, a 3 move allowance (including the last known Switch closure, one intermediate switch location, and the next valid switch closure) is accorded the player in applying the SUMDIF analysis. Should the player move too quickly about the board, as to cause the system insufficient time to evaluate switch input, or should the player skip over more than one necessary switch location in the path, the player is "arrested" for speeding, hence the SIREN message. The player's turn is terminated under those circumstances.

When the SUMDIF result is less than or equal to 3, the play continues and the system stores the current X coordinate in LASTX, the current Y coordinate at LASTY, the current N value at LASTN and adds 1 to the GAS register. If the value of the GAS register is 4, as judged at decision box 375, the system will again determine the identity of the player and access the appropriate PUTT message for that player while awaiting new input, as discussed with reference to boxes 383-385 and 314 above. If the GAS value is other than 4, the game progression circuitry puts a zero value into the GAS register at 376, determines who the player is, at 377, subtracts 1 from the PUTT register of the appropriate player, at either 381 or 378, and determines if the value in the PUTT register is zero, at 382 and 379.

As the player moves along the gameboard, the number of switch closures is tracked and the GAS and PUTT registers decremented accordingly. As noted above, at the start of each turn, the GAS variable is set at zero. Each time a player's move is confirmed, the GAS value is increased by 1. When GAS reaches the value of 4, PUTT is decreased by 1 and the GAS variable is reset to zero. The decrease in the PUTT value causes a slower "putt putt" message to be played. If the player does
not replenish his fuel supply, either by coming to the end of a turn prior to running out of gas or by driving to the gas station icon at switch 5, the system accesses audible STOP and DEAD messages and the player's turn ends. Driving to the gas station at switch results in the player's GAS register being reset to zero, generation of a gas station "Dinging" message, and a return to the fast "putt-putt" sound.

When the value in the PUTT register is zero as determined at 382 or 379, indicating that the taxi has "run out of gas," the system stops play and generates a DEAD message at 380, then follows the process flow utilized at the end of a player's turn, as described above with reference to boxes 353 through 356 and return to box 301. If the PUTT value is greater than zero, however, the system will, once again, determine the identity of the player and access the appropriate PUTT message for that player while awaiting new input, as discussed with reference to boxes 383-385 and 314, above.

As mentioned earlier, when the calculated N value is found in the table which provides storage for the destination locations, a different process flow (the so-called STOR process flow) is executed, moving from box 325 to box 400 of Figure 8-3. At the even numbered boxes 400 through 420, another series of analyses are performed to ascertain where the current N location is in relation to the last switch input, based on a check of certain LASTX or LASTY values. Each of the even-numbered decision boxes from 400-420 addresses a different identified N switch location and determines if the player had, immediately previously based upon either LASTX or LASTY, been at a roadway location adjacent to the identified N switch from which it would be permissible to enter the destination. If "No" decisions are reached at all of the even numbered decision boxes 400-420, the system, at box 422, stops play, plays the BRAKE message and the CRASH message, puts a 1 value into
the KILL register, and puts a 0 into the STEP register indicating the end of the player's turn due to execution of an impermissible move. Upon identification of the player, at 423, the system puts a zero value into the appropriate players LIV register at 424 or 425 and returns to box 301.

On the other hand, if a "Yes" decision is reached at any of the even numbered decision boxes from 400-420, the system performs the indicated WHICH register update as depicted in the associated odd numbered boxes 401-401 and proceeds to the step indicated at box 426 for identification of the destination found at the calculated location.

At box 426, the play is stopped and the BRAKE message is sounded. If this is the first step of the player's turn, as determined at box 427 by checking the value in the STEP register (where STEP=0), the game progression circuitry determines if the player selected location (WHICH) is the target location by comparing the respective N values, at 428. If the player selected location is the target location, then, at 432, the system put a 1 into the STEP register, thereby indicating that the player has successfully completed the first step of his turn and that the system will access the second destination, as analyzed in the IDLE process flow at boxes 329 through 331. Further at box 432, the game progression circuitry accesses and plays one of the two messages which convey that the player has successfully located the first destination and further plays one of the second step messages prompting the player to bring the taxi from the first destination to the randomly selected and saved second destination. Progressing along to box 454, the system puts a 1 value into the FRESH register, stores the X, Y and N values for the current location in the LASTX and LASTN registers, and returns to await input at 314.

If the player has provided switch input at the
wrong house (WHICH does not equal HOUSE), the game progression circuitry checks the value of N (or WHICH), at 429. When N is less than eleven, the system knows that the player has arrived at a house destination, even though it is the wrong destination. Therefore, at box 431, a WRONG HOUSE message is played and the system returns to await new input, after having executed the aforementioned updating at box 454. Should N be greater than eleven, the system knows that the player has arrived at a store destination which is not the target destination, the WRONG STORE" message is played at box 430, and the system again returns to await new input, after having executed the aforementioned updating at box 454.

If the player is on the second step of their turn, as determined at box 427 where the STEP value is not zero, the game progression circuitry determines if the player selected store destination (WHICH) is the target destination (STORE) at box 433, again by comparing the N values for each. A "No" determination at 433 results in a parallel response in boxes 451-453 as was provided in boxes 429-431 above. The system checks to see if the player selected destination is a store or a house and generates the appropriate "WRONG..." message.

A "Yes" determination means successful completion of the player's turn. At box 444, therefore, the game progression circuitry puts a zero back into the STEP register, indicating that the next turn will be the first step for the player, puts a 1 into the KILL register, plays one of the two messages indicating that the player successfully completed the step of finding the store (e.g., "THANK YOU FOR THE RIDE TO THE PET SHOP"), and then plays a message indicating that the player successfully completed a full turn. The system checks at 445 to determine which player successfully completed that turn, increments the SCOR register for
that player by adding 1 to the total, and then generates a message based upon that score (e.g., "YOU'VE HAD --ONE, TWO, ETC._ CUSTOMER(S)"). If the value of the SCOR register is less than 5, the system performs the updates at 454 and returns to 301, including playing the ringing telephone message.

When the value of either SCOR register equals 5, as checked at decision box 448, the game has been won and, at 449, the game progression circuitry plays a "WIN" message and the TAXI game theme and shuts down until a player again activates the game by depressing the PLAY button.

**THE "FLY" GAME**

In the FLY game, the player attempts to locate the randomly selected target (i.e., the fly). Hints are given as to the relative closeness of the player-selected switch to the target location. However, the target location may be changed during the game play (i.e., the fly may move).

Upon initialization for the FLY game by player depression of the PLAY button, the microprocessor starts the timer, puts a random value from 1-6 into the XT register for the X coordinate of the target location, puts a random value from 1-6 into the YT register for the Y coordinate of the target location, puts a zero into a TURN register to indicate that it is the player's first turn, puts a zero into the CRASH register, and starts polling the pins P1-P10 for player switch input, at 501. If there is no input, as determined at decision box 502, the timing function operates in a similar manner as in the other embodiments, except that a dual time period is demonstrated in the process flow. If a first time period has elapsed, as judged at box 503, the system inquires as to whether this is the first time through at box 504. If it is the first time through the process flow, an additional prompting message will be generated at 505 ("NYAA"). At the expiration of the
second time period at 506, a "YOU LOSE" message is generated at 507 and the system shuts down at 508.

When input is received, the game progression circuitry, at 509, stores the X and Y values for the player selected location, adds 1 to a TURN register, plays a "SWAT" message and performs a Sum:Difference calculation, as was done in the TAXI game. The SUMDIFF result is judged in determining the response to the player switch input. If the SUMDIFF is zero, as determined at decision box 510, meaning that the player has successfully located the target, the system plays a "WIN" message at 511 and shuts down at 528 until the PLAY button is again depressed.

If the SUMDIFF is not equal to zero, the game progression circuitry checks if the switch input was in a CRASH area, at 512. The CRASH areas are switch locations for which the overlay includes a picture of something breakable. If the X,Y of the input indicates a CRASH area, then the CRASH message is played at 513. Thereafter, a 1 is added to the CRASH register at 514, which keeps track of unsuccessful turns.

When the SUMDIFF does not equal zero and the input is not a CRASH area, the game progression circuitry next determines, in box 517, if the X,Y of the switch input is in a YELP area, at a switch having a picture of a dog on the overlay. If the switch input is in a YELP area, the "YELP" message is played at 518 and a 1 added to the CRASH register at 514.

In the instance when the SUMDIFF is not equal to zero, and the X,Y is in neither the CRASH nor the YELP area, the system next checks, at 519, to determine if the X,Y of the switch input is in the CAT area, at a switch having a picture of a cat on the overlay. Should X,Y be in the CAT area, the CAT message is sounded at 520 and a 1 is added to the CRASH register at box 514.

The game progression circuitry checks the value of the CRASH register at decision box 515 after every
step which a 1 is added to the register. If the value of the CRASH register is greater than 5, the system plays a "YOU LOSE" message at 516 and shuts down at 528. If the CRASH register has a value less than 5, the system next checks to see how many turns have been taken by the player, at 521. If more than a preset number (e.g., 10) turns have been taken without locating the target, the system plays the "YOU LOSE" message at 516 and shuts down at 528.

When the results of decision boxes 510, 512, 517 and 519 are all "No," or when the CRASH register has a value less than 5, the system checks to determine if more than 10 turns have been taken at 521, as above. If fewer than 10 turns have been completed, the game progression circuitry further analyzes the SUMDIF value. First, at 522, it is decided whether the SUMDIF is equal to 1. If the answer is "Yes," a message is generated at 523 telling the player that "THAT WAS CLOSE." At this juncture, as noted in the dashed box 524, there is a 50% chance that the game progression circuitry will execute a "RESET" process flow and change the target. In so doing, a message will be played (e.g., "I'M GETTING OUT OF HERE!"), NEW VALUES FOR X and Y will be placed in the XT and YT registers, 3 will be subtracted from the CRASH and TURN registers, and the system will await new input at 502.

If the SUMDIF is not 1, the game progression circuitry next asks if the SUMDIF is 2, at box 525. If the SUMDIF is 2, a MISS message is played at 527, indicating to the player that the target was "just missed" and that they are in the right section of the matrix. The system returns to await the next input at 502. However, if the SUMDIF is none of the values 0, 1 or 2, a FAR message is played at 526, indicating that the player's input is distant from the target location (e.g., "YOU'RE NOT EVEN CLOSE") and the system again waits for the next player input at 502.
Each of the embodiments of the invention have been described with reference to their respective process flow charts. Clearly, one having skill in the art can modify some of the specific steps without departing from the spirit and scope of the appended claims.
WHAT IS CLAIMED IS:

1. An electronic game apparatus for use by at least one player comprising:
   a plurality of touch sensitive input switches arranged in a matrix array for receiving Switch input from said at least one player;
   game controller means comprising a plurality of temporary storage locations, at least one permanent storage location for storing sound bit sequences, input interpretation means connected to said plurality of switches for determining the X and Y coordinates of said player switch input and for generating game input based thereon, and game progression circuitry for receiving said game input from said input interpretation circuitry, for determining game progress based thereon, and for accessing said sound bit sequences in response to said game progress determination;
   power source means for providing power for activation of said game controller means;
   activation switch means for connecting said power source means to said game controller means; and
   speaker means connected to receive said accessed sound bit sequences and reproduce sound based thereon.

2. The apparatus of Claim 1 wherein said game controller means further comprises timer means for deactivating the game controller means upon expiration of a preset time period between successive received player switch inputs.

3. The apparatus of Claim 1 wherein said game controller means includes means for randomly selecting one of said plurality of switches having target X and Y coordinates as a game target location.

4. The apparatus of Claim 3 wherein said plurality of temporary storage locations comprises means
for storing the target X and Y coordinates of said game
target location.

5. The apparatus of Claim 4 wherein said
plurality of temporary storage locations additionally
comprises means for storing the X and Y coordinates of
said player switch input as first player coordinates and
wherein said game progression circuitry additionally
comprises means for comparing first player X and Y
coordinates to successive X and Y coordinates for
successive player switch input.

6. The apparatus of Claim 4 wherein said game
progression circuitry includes means for comparing said
stored target X and Y coordinates of said game target
location with said input X and Y coordinates of said
player switch input.

7. The apparatus of Claim 6 wherein said
means for comparing further comprises means for
calculating the relative proximity of said input X and Y
coordinates to the stored target X and Y coordinates and
for producing a proximity output based on said
calculating.

8. The apparatus of Claim 3 wherein said game
controller means further comprises means for calculating
a target switch location value from said target X and Y
coordinates and for calculating a player input switch
location value from the X and Y coordinates of said
player switch input.

9. The apparatus of Claim 8 wherein said
plurality of temporary storage locations comprises means
for storing said target switch location value.

10. The apparatus of Claim 9 wherein said game
progression circuitry includes means for comparing said
stored target switch location value to said player input
switch location value and for producing a comparison
output based thereon.

11. The apparatus of Claim 3 wherein said
plurality of input switches is arrayed in an M by N
matrix, where M and N are integers and M times N equals at least two, said apparatus further comprising a graphic overlay having M times N pictured items, one pictured item overlaying each of said input switches.

12. The apparatus of Claim 11 wherein said at least one permanent storage location comprises at least one look-up table containing at least one numerical attribute value for at least one of said pictured items and wherein said plurality of temporary storage locations includes temporary look-up table means for storing at least one numerical attribute value for said game target location.

13. The apparatus of Claim 12 wherein said game progression circuitry further comprises means for retrieving said at least one numerical attribute value for the pictured item corresponding to said player switch input and means for the comparing said at least one numerical attribute value for said pictured item corresponding to said player switch input to said at least one numerical attribute value for said game target locations, said means for comparing producing a comparison output based thereon.

14. The apparatus of Claim 13 wherein said stored sound bit sequences include a plurality of attribute clue sequences and wherein said game progression circuitry accesses at least one of said plurality of attribute clue sequences in response to said comparison output from said means for comparing and provides said attribute sequences for output at said speaker means.

15. The apparatus of Claim 12 wherein said stored sound bit sequences include a plurality of attribute clue sequences and wherein said game progression circuitry accesses at least one of said plurality of attribute clue sequences in response to player switch input and provides said attribute sequences for output at said speaker means.
16. The apparatus of Claim 10 wherein said stored sound bit sequences include a plurality of location clue sequences and wherein said game progression circuitry accesses at least one of said location clue sequences in response to said comparison output.

17. The apparatus of Claim 7 wherein said stored sound bit sequences include a plurality of proximity clue sequences and wherein said game progression circuitry accesses at least one of said proximity clue sequences in response to said proximity output.

18. The apparatus of Claim 1 wherein said game progression circuitry further comprises means for incrementing a counter value for each player switch input.

19. The apparatus of Claim 18 wherein said means for incrementing a counter value is adapted to produce a counter output when a present counter value is attained.

20. The apparatus of Claim 18 wherein said means for incrementing a counter value further comprises counter reset means for changing the counter value based upon the X and Y coordinates of said player switch input.

21. The apparatus of Claim 1 wherein said stored sound bit sequences include a plurality of noise sequences for predetermined X and Y coordinates and wherein said game progression circuitry accesses at least one of said plurality of noise sequences based upon player switch input to one of said predetermined X and Y coordinates and provides said noise sequences for output at said speaker means.

22. An electronic game apparatus for use by at least one player comprising:

a plurality of touch sensitive input switches for receiving switch input from said at least one player;
game controller means comprising a plurality of temporary storage locations at least one permanent storage location for storing sound bit sequences, input interpretation means connected to said plurality of switches for determining the player switch receiving said switch input and for generating game input signals based thereon, game progression circuitry for receiving said game input signals from said input interpretation circuitry, for determining game progress based thereon, and for accessing said sound bit sequences in response to said game progress determination, and means for randomly selecting one of said input switches as a target switch;

power source means for providing power for activation of said game controller means;

activation switch means for connecting said power source means to said game controller means; and

speaker means connected to receive said accessed sound bit sequences and reproduce sound based thereon.

23. The apparatus of Claim 22 wherein said at least one permanent storage location comprises at least one look-up table containing at least one numerical attribute value for each of said plurality of input switches, wherein said plurality of temporary storage locations includes temporary look-up table means for storing at least one numerical attribute value for said target switch, and wherein said game progression circuitry further comprises means for retrieving said at least one numerical attribute value for the player switch and means for the comparing said at least one numerical attribute value for said player switch to said at least one numerical attribute value for said target switch.

24. The apparatus of Claim 22 wherein said game progression circuitry further comprises means for
determining the relative proximity of said player switch to said target switch.

25. The apparatus of Claim 22 wherein said game progression circuitry further comprises means for comparing said player switch to said target switch.

26. A method for implementing an electronic game for use by at least one player, said game including a plurality of input switches arranged in a matrix array for receiving switch input from said player and game controller means comprising the steps of:
   receiving player switch input to one of said plurality of input switches;
   determining at said game controller means if said player switch input is valid; and
   generating game progress responses based on said valid player switch input.

27. The method of Claim 26 further comprising the step of deriving the X and Y coordinates of said player switch input.

28. The method of Claim 26 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, wherein the step of generating game progress responses includes the steps of:
   accessing at least one of said plurality of sound bit sequences; and
   outputting said accessed sound bit sequences at said speaker means.

29. The method of Claim 26 wherein said game controller means further comprises timer means and further comprising the steps of:
   counting time between player switch input;
   comparing said counted time to a preset time period; and
   deactivating said game controller means when said counted time equals said preset time period.
30. The method of Claim 26 further comprising the step of randomly selecting one of said matrix locations as a game target location.

31. The method of Claim 30 wherein said game controller means further comprises a plurality of temporary storage locations and further comprising the step of:

storing the X and Y coordinates of said game target location at one of said plurality of temporary storage locations.

32. The method of Claim 31 wherein said generating comprises comparing said stored X and Y coordinates of said game target location with said input X and Y coordinates of said player switch input and producing game progress responses based thereon.

33. The method of Claim 31 wherein said generating comprises:

- calculating the relative proximity of said input X and Y coordinates to the target X and Y coordinates; and

- producing game progress responses based thereon.

34. The method of Claim 30 further comprising the steps of:

- calculating a target switch location value from said target X and Y coordinates; and

- calculating a player input switch location value from said player switch input.

35. The method of Claim 34 wherein said game further comprises at least one temporary storage location associated with said game controller means and further comprising the step of:

storing said target switch location value at said at least one temporary storage location.

36. The method of Claim 35 further comprising the steps of:
comparing said target switch location value to said player input switch location value; and producing a comparison output based upon said comparing.

37. The method of Claim 26 wherein said matrix array is an M by N array, where M and N are integers and M times N equals at least two, said apparatus further comprising a graphic overlay having M times N pictured items, one pictured item overlaying each of said input switches and wherein said game controller means further comprises at least one permanent storage location, further comprising the steps of:

storing at least one numerical attribute value for at least one of said pictured items in said at least one permanent storage location; and

storing at least one numerical attribute value for said game target location at one of said plurality of temporary storage locations.

38. The method of Claim 37 further comprising the steps of:

retrieving said at least one numerical attribute value for the pictured item corresponding to said player switch input;

comparing said at least one numerical attribute value for said pictured item corresponding to said player switch input to said at least one numerical attribute value for said game target location; and producing a comparison output based thereon.

39. The method of Claim 38 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, further comprising the steps of:

storing sound bit sequences including a plurality of attribute clue sequences at said at least one permanent storage location;
accessing at least one of said plurality of attribute clue sequences in response to said comparison output; and
outputting said accessed sound bit sequences at said speaker means.

40. The method of Claim 26 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, further comprising the steps of:
- storing sound bit sequences including a plurality of attribute clue sequences at said at least one permanent storage location;
- accessing at least one of said attribute clue sequences in response to said player switch input; and
- outputting said accessed sound bit sequences at said speaker means.

41. The method of Claim 36 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, further comprising the steps of:
- storing sound bit sequences including a plurality of location clue sequences at said at least one permanent storage location;
- accessing at least one of said location clue sequences in response to said comparison output; and
- outputting said accessed sound bit sequences at said speaker means.

42. The method of Claim 33 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, further comprising the steps of:
- storing sound bit sequences including a
plurality of proximity clue sequences at said at least one permanent storage location;
accessing at least one of said proximity clue sequences based on said calculating; and
outputting said accessed sound bit sequences at said speaker means.

43. The method of Claim 26 wherein said game controller means further comprises counter means and wherein said generating additionally comprises the step of:

incrementing said counter means by a value of one for each player switch input.

44. The method of Claim 43 further comprising the step of producing a counter output when said counter value reaches a preset limit.

45. The method of Claim 27 wherein said game controller means further comprises counter means and wherein said generating additionally comprises the step of:

incrementing said counter means by a value of one for each player switch input.

46. The method of Claim 45 further comprising the step of producing a counter output when said counter value reaches a preset limit.

47. The method of Claim 46 further comprising the step of changing the counter value based upon said deriving said X and Y coordinates.

48. The method of Claim 27 wherein said electronic game further comprises speaker means and wherein said game controller means includes at least one permanent storage location for storing a plurality of sound bit sequences, further comprising the steps of:

storing sound bit sequences including a plurality of noise sequences at said at least one permanent storage location;
accessing at least one of said noise sequences in response to said deriving of X and Y
coordinates of said player switch input; and
outputting said accessed sound bit
sequences at said speaker means.

49. The method of Claim 30 further wherein
said game controller means further comprises at least
one permanent storage location, further comprising the
steps of:

storing at least one numerical attribute
value for each of said input switches in said at least
one permanent storage location;
storing at least one numerical attribute
value for said game target location at one of said
plurality of temporary storage locations;
retrieving said at least one numerical
attribute value for said player switch input; and
comparing said at least one numerical
attribute value for said player switch input to said at
least one numerical attribute value for said game target
location.

50. The method of Claim 30 further comprising
the step of comparing said player switch input to said
game target location.

51. The method of Claim 26 wherein said
electronic game further comprises speaker means and
wherein said game controller means includes at least one
permanent storage location for storing a plurality of
sound bit sequences, the method further comprising the
steps of:

randomly selecting one of said matrix
locations as a game target location;
accessing at least one of said plurality
of sound bit sequences relating to said game target
location; and
outputting said accessed sound bit
sequences at said speaker means.

52. The method of Claim 51 wherein said
receiving player switch input comprises:
player actuating switches intermediate a
starting switch location and said game target location
by moving a finger along said switches to produce player
input signals; and

5

receiving said player input signals at
said game controller means.

53. The method of Claim 51 wherein said
receiving player switch input comprises:

player actuating switches intermediate a
starting switch location and said game target locations
by moving a game piece along said switches to produce
player input signals; and

receiving said player input signals at
said game controller means.

15
FIG. 2

SUBSTITUTE SHEET (RULE 26)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>HELP = 0</th>
<th>HELP = 1</th>
<th>HELP = 2</th>
<th>HELP = 3</th>
<th>HELP = 4</th>
<th>HELP = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MOUSE</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
<tr>
<td>2</td>
<td>DUCK</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
<tr>
<td>3</td>
<td>DOG</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
<tr>
<td>4</td>
<td>CAT</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
<tr>
<td>5</td>
<td>CAR</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
<tr>
<td>6</td>
<td>HOUSE</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 2</td>
<td>HELP = 3</td>
<td>HELP = 4</td>
<td>HELP = 5</td>
</tr>
</tbody>
</table>

FIG. 5-1

SUBSTITUTE SHEET (RULE 26)
<table>
<thead>
<tr>
<th>24 CANARY</th>
<th>23 SHIP</th>
<th>22 GUITAR</th>
<th>21 PHONE</th>
<th>20 BURGER</th>
<th>19 CHAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG 7</td>
<td>BIG 13</td>
<td>BIG 6</td>
<td>BIG 4</td>
<td>BIG 3</td>
<td>BIG 8</td>
</tr>
<tr>
<td>LOUD 3</td>
<td>LOUD 5</td>
<td>LOUD 4</td>
<td>LOUD 1</td>
<td>LOUD 0</td>
<td>LOUD 0</td>
</tr>
<tr>
<td>HARD 0</td>
<td>HARD 1</td>
<td>HARD 1</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
</tr>
<tr>
<td>COLR 5</td>
<td>COLR 1</td>
<td>COLR 4</td>
<td>COLR 5</td>
<td>COLR 0</td>
<td>COLR 3</td>
</tr>
<tr>
<td>FAST 2</td>
<td>FAST 4</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
</tr>
<tr>
<td>HELP = 5</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>30 KEY</th>
<th>29 ELEPHANT</th>
<th>28 BALLOON</th>
<th>27 SPOON</th>
<th>26 BED</th>
<th>25 SNOWFLAKE</th>
<th>32 BUTTERFLY</th>
<th>31 DRUM</th>
<th>34 HANGER</th>
<th>33 CAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG 1</td>
<td>BIG 10</td>
<td>BIG 5</td>
<td>BIG 2</td>
<td>BIG 0</td>
<td>BIG 0</td>
<td>BIG 1</td>
<td>BIG 5</td>
<td>BIG 5</td>
<td>BIG 5</td>
</tr>
<tr>
<td>LOUD 0</td>
<td>LOUD 5</td>
<td>LOUD 0</td>
<td>LOUD 1</td>
<td>LOUD 1</td>
<td>LOUD 0</td>
<td>LOUD 0</td>
<td>LOUD 1</td>
<td>LOUD 0</td>
<td>LOUD 0</td>
</tr>
<tr>
<td>HARD 1</td>
<td>HARD 0</td>
<td>HARD 1</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
<td>HARD 0</td>
</tr>
<tr>
<td>COLR 5</td>
<td>COLR 4</td>
<td>COLR 5</td>
<td>COLR 5</td>
<td>COLR 1</td>
<td>COLR 4</td>
<td>COLR 1</td>
<td>COLR 4</td>
<td>COLR 0</td>
<td>COLR 0</td>
</tr>
<tr>
<td>FAST 3</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 2</td>
<td>FAST 0</td>
<td>FAST 0</td>
<td>FAST 0</td>
</tr>
<tr>
<td>HELP = 0</td>
<td>HELP = 2</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 0</td>
<td>HELP = 1</td>
<td>HELP = 0</td>
<td>HELP = 5</td>
<td>HELP = 4</td>
</tr>
</tbody>
</table>

**FIG. 5-2**

SUBSTITUTE SHEET (RULE 26)
Fig. 6-2

From 210
Fig. 6-1

N = TARG

203

YES

Colr X
TColr = 0

NO

218

219

PUT RANDOM
( 2 - 5 )
INTO R

222

220

YES

PUT 2 INTO
CLU

221

NO

PUT 1 INTO
CLU

227

PUT 3 INTO
CLU

229

PUT 4 INTO
CLU

228

YES

BIG > TBIG

NO

230

PUT 5 INTO
CLU

231

NO

LOUD < TLOUD

YES

234

PUT 7 INTO
CLU

233

NO

LOUD > TLOUD

232

PUT 6 INTO
CLU

226

YES

R = 2

NO

223

R = 3

239

PUT 2 INTO R

234

TO 225 Fig. 6-3

SUBSTITUTE SHEET (RULE 26)
FROM 224 FIG. 6-2

225

R = 4

235

HARD > THARD

YES

NO

236

PUT 8 INTO CLU

237

HARD < THARD

YES

NO

238

PUT 9 INTO CLU

239

240

FAST > TFAST

YES

NO

241

PUT 10 INTO CLU

242

FAST < TFAST

YES

NO

243

PUT 11 INTO CLU

244

ADD 10 TO CLU

SAY "IM", ( )"CLU

245

CLU < 14

YES

NO

246

SAY "THAT"

247

SAY "FILL"

EXIT

TO 202

FIG 6-3

SUBSTITUTE SHEET (RULE 26)
FROM 250 FIG. 6-1

ENTER

PUT RANDOM (4) INTO R

252

TCOLR = 0 ?

YES

HELP=0 ?

NO

R = 4 ?

YES

NO

WHAT IS THE VALUE OF HELP?

HELP=1

HELP=2

HELP=5

HELP=3

HELP=4

256

257

258

259

260

261

R > 2

YES

NO

262

263

264

265

266

267

268

269

270

271

SAY IM

SAY I HAVE

SAY HELP1

SAY HELP2

SAY HELP3

SAY HELP4

SAY COLOR TCOLR

EXIT

TO 202

FIG. 6-4

SUBSTITUTE SHEET (RULE 26)
INITIALIZATION

PUT 0 INTO ALL VARIABLES
PLAY "THEME"
START TIMER
PUT 1 INTO KILL

NEW INPUT

T1 > LIMIT?

PLAY "BEEP"

SHUT DOWN

X + Y = 2?

0

WHO = 2

PUT 1 INTO WHO

PLAY "BEEP"
GO TO SUBROUTINE "CLIENT"

X + Y = 12?

0

WHO = 1

KILL = 1?

TO: FIG. 8-2

FIG. 8-1

SUBSTITUTE SHEET (RULE 26)
FROM FIG. 8-1

LOAD X, Y
PUT (X+6Y-6) INTO N

323

PUT 2 INTO WHO

328

N IS IN TABLE (STOR)?

324

YES

325

FRESH=1?

326

NO

YES

PUT X INTO LASTX
PUT Y INTO LASTY
PUT N INTO LASTN

GO TO SUBROUTINE *STOR*
FIGS. 8-3 THROUGH 8-6

NO

STEP=0?

329

330

PLAY "1ST GO (1, 2, OR 3), HOUSE"

YES

PLAY "2ND GO (1, 2, OR 3), STORE"

331

PLAY "BEEP"
GO TO SUBROUTINE *CLIENT*
FIGS. 8-11

GO TO SUBROUTINE *MOVER*
FIGS. 8-7 THROUGH 8-10

TO 301

FIG. 8-2
SUBSTITUTE SHEET (RULE 26)
FROM 325

ENTER

400
N = 10
AND LASTX = 3
?

402
N = 11
AND LASTX = 6
?

404
N = 16
AND LASTX = 3
?

406
N = 17
AND LASTX = 6
?

408
N = 23
AND LASTX = 6
?

410
N = 2
AND LASTY = 2
?

412
N = 35
AND LASTY = 5
?

401
PUT 1 INTO WHICH

403
PUT 2 INTO WHICH

405
PUT 3 INTO WHICH

407
PUT 4 INTO WHICH

409
PUT 5 INTO WHICH

411
PUT 11 INTO WHICH

413
PUT 12 INTO WHICH

TO FIG 8-4

FIG. 8-3

SUBSTITUTE SHEET (RULE 26)
FROM FIG 8 - 3

16/26

N = 14
AND LASTX = 1
?
YES
NO

N = 20
AND LASTX = 1
?
YES
NO

N = 26
AND LASTY = 6
?
YES
NO

N = 27
AND LASTY = 6
?
YES
NO

PLAY STOP
PLAY "BRAKE"
PLAY "CRASH"
PUT 1 INTO KILL
PUT 0 INTO STEP

WHO = 1
?
YES

PUT 0 INTO 2LIV

PUT 0 INTO 1LIV

EXIT

TO 301

FIG. 8-4
FROM FIG 8-4

PLAY STOP
PLAY "BRAKE"

STEP = 0

NO

YES

WHICH=HOUSE

PUT 1 INTO STEP
PLAY "$RHOUSE$
( 1 OR 2 )
PLAY "$2ND GO$
( 1, 2, OR 3 )
PLAY "$STORE$"

PLAY "$RONG2$"
"(WHICH) STORE"

 WHICH<11

NO

YES

PLAY "$RONG1$"
"(WHICH) HOUSE"

 WHICH=STORE

PUT 0 INTO STEP
PLAY "$RSTOR(1 OR 2)$"
PLAY "WHICH"
PUT 1 INTO KILL
PLAY "OBOY"

WHO=1

NO

YES

ADD 1 TO $2SCOR$
PLAY "$2SCORSCOR$"

TO FIG. 8-6

FIG. 8-5
SUBSTITUTE SHEET (RULE 26)
FIG. 8-6

SUBSTITUTE SHEET (RULE 26)
FROM 324 FIG. 8-5

350

\[ N=5 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

357

\[ N=3 \, \text{AND} \, \text{LASTN}=2 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

358

\[ N=4 \, \text{AND} \, \text{LASTN}=10 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

359

\[ N=22 \, \text{AND} \, \text{LASTN}=16 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

360

\[ N=29 \, \text{AND} \, \text{LASTN}=23 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

361

\[ N=34 \, \text{AND} \, \text{LASTN}=35 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

362

\[ N=33 \, \text{AND} \, \text{LASTN}=27 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

363

\[ N=21 \, \text{AND} \, \text{LASTN}=27 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

351

\[ \text{LASTN}=11 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

349

\[ \text{FRESH} = 1 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

348

PLAY STOP
PLAY "DING"
PUT 0 INTO GAS
PUT 1 INTO FRESH
PUT N INTO LASTN
PUT X INTO LASTX
PUT Y INTO LASTY

347

\[ \text{WHO} = 1 \, ? \]
\[ \text{YES} \]
\[ \text{NO} \]

346

PUT 5 INTO 1 PUTT
PLAY "( ) 1 PUTT"

345

PUT 5 INTO 2 PUTT
PLAY "( ) 2 PUTT"

EXIT

TO 364 FIG. 8-8

TO 352 FIG. 8-8

Figure 8-7
FROM 367 FIG. 8-8

☐ FRESH = 1

☐ PUT abs (X - LASTX) + abs (Y - LASTY) INTO SUMDIF

☐ SUMDIF = 0

☐ SUMDIF ≤ 3

☐ PUT X INTO LASTX
☐ PUT Y INTO LASTY
☐ PUT N INTO LASTN
☐ PUT O INTO FRESH

☐ GAS = 4

☐ PUT 0 INTO GAS

☐ WHO = 1

☐ PLAY STOP
☐ PLAY "SIREN"

TO FIG. 8-10

FIG. 8-9
SUBSTITUTE SHEET (RULE 26)
FROM 301 → TAXI

SWITCH INPUT

A7 / OR 36

YES

WHO = 1?

YES

1LIV=0

YES

1PUTT<3

PUTS 5 INTO 1PUTT

NO

2LIV=0

YES

2PUTT<3

PUTS 5 INTO 2PUTT

NO

2PUTT<3

PUT 1 INTO LASTX

PUT 1 INTO LASY

PUT 1 INTO LASTN

PUT RANDOM (1-5) INTO HOUSE

PUT RANDOM (11-16) INTO STORE

PUT 0 INTO SUMDIF

PUT 0 INTO KILL

PUT 0 INTO STEP

PUT 1 INTO FRESH

PLAY "1 ST GO (1, 2 OR 3)"

50%—

PLAY "( ) HOUSE"

50%

PLAY "( +5) HOUSE"

TO 301 AND 314

FIG. 8-11

SUBSTITUTE SHEET (RULE 26)
START TIMER 6
PUT RANDOM (6) INTO XT
PUT RANDOM (6) INTO YT
PUT 0 INTO TURN
PUT 0 INTO CRASH

NO

NEW INPUT

T1>LIMIT

YES

FIRST TIME THROUGH?

PLAY "NYAA"

T2>LIMIT?

YES

PLAY "LOSE"

NO

SHUT DOWN

NO

STORE X
STORE Y
ADD 1 TO TURN
PLAY "SWAT"
PUT 1X-XT1+Y-YT1 INTO SUMDIF

NO

SUMDIF=0?

YES

PLAY "WIN"

NO

X, Y IN (CRASH)?

PLAY "CRASH"

NO

X, Y IN (YELP)?

PLAY "YELP"

NO
FROM FIG. 10-1

X, Y IN (CAT) ?

519

YES

520

PLAY "CAT"

514

ADD 1 TO CRASH

521

TURN >10 ?

YES

522

CRASH >5?

515

YES

516

PLAY "LOSE"

523

PLAY "CLOSE"

524

SUMDIF=1

NO

525

PLAY "MISS"

527

526

PLAY "FAR"

528

SHUT DOWN

(50% CHANCE)
PLAY "RESET"
PUT RANDOM(6) INTO XT
PUT RANDOM(6) INTO YT
SUBTRACT 3 FROM TURN (LIMIT 0)
SUBTRACT 3 FROM CRASH (LIMIT 0)

FIG. 10-2

SUBSTITUTE SHEET (RULE 26)
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) : A63F 7/06, 9/00, 9/18, 9/22; G06F 17/00; G06G 7/48
US CL : 364/410; 463/7, 9, 15, 31, 35, 37
According to International Patent Classification (IPC) or to both national classification and IPC.

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US, A, 4,320,901 (MORRISON ET AL.) 23 March 1982, see column 3 line 34 to column 4 line 16, Figs. 1-5, and Games 1-4.</td>
<td>1-7, 17-22, 24-33, 40, 42-48, 50-52</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>8-16, 23, 34-39, 41, 49, 53</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. [ ]

See patent family annex.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be part of particular relevance
  "E" earlier document published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

& document member of the same patent family

Date of the actual completion of the international search
26 APRIL 1996

Date of mailing of the international search report
21 MAY 1996

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
MARK SAGER

Telephone No. (703) 308-0785

Form PCT/ISA/210 (second sheet)(July 1992)*
### INTERNATIONAL SEARCH REPORT

**C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US, A, 5,411,271 (MIRANDO) 02 May 1995, see column 3 line 12 to column 8 line 41, and Figs. 1-4H.</td>
<td>1-7, 17-22, 24-33, 40, 42-48, 50-52</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>8-16, 23, 34-39, 41, 49, 53</td>
</tr>
<tr>
<td>Y</td>
<td>US, 3,042,408 (K. G. JOHNSON) 03 July 1962, see column 2 line 40 to column 3 line 2, and lines 64-68; and Figs. 1-141 especially Figs. 2-16.</td>
<td>11-15, 23, 37, 38</td>
</tr>
</tbody>
</table>
B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

APS

Search Terms: game or demonstration or teach or learn, location or address or XY coordinate or X and Y coordinate, time or timer or timed, touch or push or select or selection or selectable, sound or music or sound bits, and processor or microprocessor or controller.