Characters that form a word or a sentence having a meaning are assigned to squares, and are adjacently disposed so that a word or sentence is reproduced by reading the characters along the right route. The player inputs an answer route by tracing the squares with one stroke so as to read the characters correctly. A route indicator (32) is displayed along the answer route, and a moving object (34) is moved along the route indicator (32).
FIG. 1
FIG. 2

STARTS WITH NAME OF JUICE BEVERAGE

HINT NO. 2

HINT NO. 3

LEVEL 2  20 POINTS  2 minutes

TRACE THE ANSWER

f a n
s a t
i c

DISPLAY HINT  FINISHED
FIG. 3

1. Trace the answer:

   - Fan
   - Sat
   - Tic

2. Display hint

3. Finished
FIG. 4

(1)

W10 → 10

TRACE THE ANSWER

14  

34

1408

12

32

20

(2)

W12 → 10

TRACE THE ANSWER

1408

20

(3)

W14 → 10

TREASURE FOUND!

1408

36

12

38

NEXT QUESTION

FINISHED

16

FINISHED

40
<table>
<thead>
<tr>
<th>Function</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING SECTION</td>
<td></td>
</tr>
<tr>
<td>GAME CALCULATION SECTION</td>
<td></td>
</tr>
<tr>
<td>RIGHT CHARACTER STRING DETERMINATION SECTION</td>
<td></td>
</tr>
<tr>
<td>DIFFICULTY LEVEL SETTING SECTION</td>
<td></td>
</tr>
<tr>
<td>QUESTION ARRAY GENERATION SECTION</td>
<td></td>
</tr>
<tr>
<td>HINT DISPLAY CONTROL SECTION</td>
<td></td>
</tr>
<tr>
<td>ANSWER ROUTE MANAGEMENT SECTION</td>
<td></td>
</tr>
<tr>
<td>FILTER EFFECT CONTROL SECTION</td>
<td></td>
</tr>
<tr>
<td>SOUND GENERATION SECTION</td>
<td></td>
</tr>
<tr>
<td>IMAGE GENERATION SECTION</td>
<td></td>
</tr>
<tr>
<td>COMMUNICATION CONTROL SECTION</td>
<td></td>
</tr>
<tr>
<td>STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SYSTEM PROGRAM</td>
<td></td>
</tr>
<tr>
<td>GAME PROGRAM</td>
<td></td>
</tr>
<tr>
<td>GAME SCREEN SETTING DATA</td>
<td></td>
</tr>
<tr>
<td>MOVING OBJECT DATA</td>
<td></td>
</tr>
<tr>
<td>SQUARE DISPLAY DATA</td>
<td></td>
</tr>
<tr>
<td>PRIZE IMAGE LIBRARY</td>
<td></td>
</tr>
<tr>
<td>QUESTION CONDITION SETTING DATA</td>
<td></td>
</tr>
<tr>
<td>RIGHT CHARACTER STRING SET</td>
<td></td>
</tr>
<tr>
<td>ARRAY MODEL DATA</td>
<td></td>
</tr>
<tr>
<td>WORD DICTIONARY DATA</td>
<td></td>
</tr>
<tr>
<td>DIFFICULTY LEVEL</td>
<td></td>
</tr>
<tr>
<td>TOTAL ACQUIRED POINTS</td>
<td></td>
</tr>
<tr>
<td>US 2011/0244937 A1</td>
<td></td>
</tr>
</tbody>
</table>

FIG. 6
FIG. 7

<table>
<thead>
<tr>
<th>CURRENTLY ACQUIRED POINT CONDITION</th>
<th>PRIZE IMAGE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1~100</td>
<td>SCATTERED COPPER COINS</td>
</tr>
<tr>
<td>101~200</td>
<td>ACCUMULATED COPPER COINS</td>
</tr>
<tr>
<td>201~300</td>
<td>SCATTERED SILVER COINS</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### QUESTION CONDITION SETTING OF EACH ITEM

<table>
<thead>
<tr>
<th>APPLIED LEVEL SETTING</th>
<th>NUMBER OF CORNERS</th>
<th>MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES</th>
<th>NUMBER OF BENDS</th>
<th>NUMBER OF LEFTWARD STRAIGHT LINES</th>
<th>BLIND BOX</th>
<th>FIRST CHARACTER CAPITALIZATION</th>
<th>UNNECESSARY CHARACTER</th>
<th>COLOR CHANGE</th>
<th>FIRST SQUARE ARRANGEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL 1</td>
<td>NOT LIMITED</td>
<td>NOT LIMITED</td>
<td>3 OR LESS</td>
<td>4 OR LESS</td>
<td>0 (NONE)</td>
<td>ON</td>
<td>0</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>LEVEL 2</td>
<td>10 OR LESS</td>
<td>3 OR MORE</td>
<td>6 OR LESS</td>
<td>6 OR LESS</td>
<td>0 (NONE)</td>
<td>OFF</td>
<td>0</td>
<td>ON</td>
<td>NOT LIMITED</td>
</tr>
<tr>
<td>LEVEL 2</td>
<td>10 OR LESS</td>
<td>4 OR LESS</td>
<td>6 OR LESS</td>
<td>6 OR LESS</td>
<td>0 (NONE)</td>
<td>ON</td>
<td>0</td>
<td>ON</td>
<td>NOT LIMITED</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>LEVEL 5</td>
<td>5 OR LESS</td>
<td>3 OR MORE</td>
<td>NOT LIMITED</td>
<td>NOT LIMITED</td>
<td>1</td>
<td>OFF</td>
<td>0</td>
<td>OFF</td>
<td>NOT LIMITED</td>
</tr>
<tr>
<td>LEVEL 5</td>
<td>5 OR LESS</td>
<td>3 OR MORE</td>
<td>NOT LIMITED</td>
<td>NOT LIMITED</td>
<td>2</td>
<td>OFF</td>
<td>1</td>
<td>OFF</td>
<td>NOT LIMITED</td>
</tr>
<tr>
<td>LEVEL 5</td>
<td>5 OR LESS</td>
<td>3 OR LESS</td>
<td>NOT LIMITED</td>
<td>NOT LIMITED</td>
<td>1</td>
<td>OFF</td>
<td>2</td>
<td>OFF</td>
<td>NOT LIMITED</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
FIG. 9

(1) NUMBER OF CORNERS: 4

(2) NUMBER OF CORNERS: 5

(3) NUMBER OF CORNERS: 8
FIG. 10

MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES

DIFFICULTY LEVEL

(1) MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES: 5

(2) MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES: 5

(3) MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES: 3
FIG. 11

(1)  
DIFFICULTY LEVEL

NUMBER OF BENDS

(2)  
NUMBER OF BENDS: 7

(3)  
NUMBER OF BENDS: 14
FIG. 12

(1) 3

0 2 4 6 10 12
NUMBER OF LEFTWARD STRAIGHT LINES

(2) RIGHT CHARACTER STRING "fantastic"

NUMBER OF LEFTWARD STRAIGHT LINES: 1

(3) RIGHT CHARACTER STRING "fantastic"

NUMBER OF LEFTWARD STRAIGHT LINES: 2

(4) RIGHT CHARACTER STRING "fantastic"

NUMBER OF LEFTWARD STRAIGHT LINES: 2
FIG. 13

RIGHT CHARACTER STRING "fantastic"  RIGHT ANSWER ROUTE  BLIND BOX: PRESENT
n a c
f f i
s t

FIG. 14

RIGHT CHARACTER STRING "wikipedia"  RIGHT ANSWER ROUTE  FIRST CHARACTER CAPITALIZATION: ON
k i a
i w i
p e d

FIG. 15

RIGHT CHARACTER STRING "fantastic", UNNECESSARY CHARACTER: PRESENT  RIGHT ANSWER ROUTE
a t n
s r a
t u f
i c
FIG. 16

RIGHT CHARACTER STRING
“UNITED STATES OF AMERICA”

WORD COLOR CHANGE: OFF

<table>
<thead>
<tr>
<th>T</th>
<th>I</th>
<th>N</th>
<th>I</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>D</td>
<td>U</td>
<td>R</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>S</td>
<td>S</td>
<td>O</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>T</td>
<td>E</td>
<td>F</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

(1)

RIGHT ANSWER ROUTE

(2)

WORD COLOR CHANGE: ON

<table>
<thead>
<tr>
<th>T</th>
<th>I</th>
<th>N</th>
<th>I</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>D</td>
<td>U</td>
<td>R</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>S</td>
<td>S</td>
<td>O</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>T</td>
<td>E</td>
<td>F</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

(3)
**FIG. 17**

<table>
<thead>
<tr>
<th>RIGHT CHARACTER STRING ID</th>
<th>RIGHT CHARACTER STRING</th>
<th>FIRST HINT</th>
<th>SECOND HINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL1001</td>
<td>TEKKAMAKI</td>
<td>JAPANESE FOOD</td>
<td>ENDS WITH &quot;KI&quot;</td>
</tr>
<tr>
<td>TL1002</td>
<td>CHOCOLATE COOKIE</td>
<td>CAKE</td>
<td>STARTS WITH &quot;OH&quot;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**EXTRACTION:**

```
RIGHT CHARACTER STRING SET

514a
APPLIED LEVEL  LEVEL 1

514b
NUMBER-OF-CHARACTER CONDITION  8 OR LESS
```

**EXTRACTION2:**

```
<table>
<thead>
<tr>
<th>514d</th>
<th>514e</th>
<th>514f</th>
<th>514g</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
FIG. 18

(1) ARRAY MODEL DATA
- ARRAY ID: MTO23
- SQUARE COUNT: 5

(2) ARRAY MODEL DATA
- ARRAY ID: MTO71
- SQUARE COUNT: 10

ARRAY MODEL

ARRAY COORDINATES
- Points: P0, P1, P2, P3, P4, P5

ARRANGEMENT ORDER
- Points: P0, P1, P2, P3, P4, P5
FIG. 19

QUESTION ARRAY MODEL

QUESTION ARRAY MODEL

CHARACTER LINK DATA

TABLE: SQUARE REPRESENTATIVE POINT COORDINATES vs. CHARACTER

<table>
<thead>
<tr>
<th>SQUARE REPRESENTATIVE POINT COORDINATES</th>
<th>CHARACTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X1,Y1)</td>
<td>a</td>
</tr>
<tr>
<td>(X2,Y1)</td>
<td>p</td>
</tr>
<tr>
<td>(X2,Y2)</td>
<td>p</td>
</tr>
<tr>
<td>(X1,Y2)</td>
<td>l</td>
</tr>
<tr>
<td>(X1,Y3)</td>
<td>e</td>
</tr>
</tbody>
</table>

FIG. 20

ANSWER ROUTE INFORMATION

TABLE: TOUCH ORDER vs. TOUCHED SQUARE REPRESENTATIVE POINT COORDINATES

<table>
<thead>
<tr>
<th>TOUCH ORDER</th>
<th>TOUCHED SQUARE REPRESENTATIVE POINT COORDINATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(X1,Y1)</td>
</tr>
<tr>
<td>2</td>
<td>(X2,Y2)</td>
</tr>
<tr>
<td>3</td>
<td>(X3,Y3)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
FIG. 23

QUESTION ARRAY GENERATION PROCESS

S30. SELECTS SETTING OF EACH ITEM OF QUESTION CONDITION BASED ON CURRENT DIFFICULTY LEVEL

S32. IS QUESTION CONDITION "UNNECESSARY CHARACTER" SET TO OTHER THAN "0" ?

S34. SELECTS ONE OF ARRAY MODEL DATA FOR WHICH SQUARE COUNT IS EQUAL TO NUMBER OF CHARACTERS OF RIGHT CHARACTER STRING

S40. DOES SELECTED ARRAY MODEL DATA SATISFY EACH OF "NUMBER OF CORNERS", "NUMBER OF BENDS", "MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES", "NUMBER OF LEFTWARD STRAIGHT LINES", AND "FIRST SQUARE ARRANGEMENT" ?

S35. UPDATES RIGHT CHARACTER STRING BY ADDING CHARACTERS IN NUMBER CORRESPONDING TO QUESTION CONDITION "UNNECESSARY CHARACTER" AT HEAD OR END OF RIGHT CHARACTER STRING

S36. SELECTS ARRAY MODEL DATA FOR WHICH SQUARE COUNT IS EQUAL TO NUMBER OF CHARACTERS OF RIGHT CHARACTER STRING THAT HAS BEEN UPDATED

S42. DISCARDS CURRENT SELECTION OF ARRAY MODEL DATA
FIG. 24

S70. IS NATURAL NUMBER SET AS QUESTION CONDITION "BLIND BOX"?

S72. HIDES CHARACTERS IN NUMBER CORRESPONDING TO NATURAL NUMBER SET AS QUESTION CONDITION "BLIND BOX"

S84. IS QUESTION CONDITION "FIRST CHARACTER CAPITALIZATION" SET TO "ON"?

S86. CAPITALIZES FIRST CHARACTER OF RIGHT CHARACTER STRING, AND UN-CAPITALIZES REMAINING CHARACTERS

S88. GENERATES QUESTION ARRAY DATA BY BLENDING CHARACTER OF RIGHT CHARACTER STRING WITH SQUARE OF ARRAY MODEL OF CURRENTLY SELECTED ARRAY MODEL DATA BASED ON ARRANGEMENT ORDER

RETURN
FIG. 25

S140
ANSWER ROUTE DISPLAY PROCESS

S142
IS TOUCHED SQUARE STARTING POINT?
YES
DISPLAYS STARTING POINT INDICATOR IN SQUARE

NO
HAS SQUARE ADJACENT TO FINAL SQUARE ALONG ANSWER ROUTE BEEN TOUCHED?

S143
NO

S144
HAS SQUARE PRECEDING FINAL SQUARE ALONG ANSWER ROUTE BEEN TOUCHED?
YES
STORES COORDINATES OF REPRESENTATIVE POINT OF TOUCHED SQUARE AS ANSWER ROUTE INFORMATION

NO
DELETE COORDINATES OF REPRESENTATIVE POINT OF TOUCHED SQUARE FROM ANSWER ROUTE INFORMATION

S146
DISPLAYS ROUTE INDICATOR SO THAT ROUTE INDICATOR EXTENDS TO REPRESENTATIVE POINT OF CURRENTLY TOUCHED SQUARE ALONG ANSWER ROUTE

S148
DISPLAYS ROUTE INDICATOR SO THAT ROUTE INDICATOR IS SHORTENED TO REPRESENTATIVE POINT OF PRECEDING SQUARE ALONG ANSWER ROUTE

S150
RETURN
FIG. 26

CLEAR EFFECT DISPLAY PROCESS

S210: DISPLAYS MOVING OBJECT ON STARTING POINT END OF ROUTE INDICATOR

S212: CHANGES DISPLAY STATE OF STARTING POINT SQUARE

S214: MOVES MOVING OBJECT ALONG ROUTE INDICATOR, AND DELETES AREA OF ROUTE INDICATOR THROUGH WHICH MOVING OBJECT HAS PASSED

S216: HAS MOVING OBJECT MOVED TO ANOTHER SQUARE?

NO

YES

S218: CHANGES DISPLAY STATE OF SQUARE

S220: HAS MOVING OBJECT REACHED END OF ROUTE INDICATOR?

NO

YES

S222: SELECTS PRIZE IMAGE CORRESPONDING TO CURRENTLY ACQUIRED POINTS

S224: CAUSES SELECTED PRIZE IMAGE TO APPEAR FROM BEHIND SQUARES FOR WHICH DISPLAY STATE HAS BEEN CHANGED

RETURN
FIG. 27

FAILURE EFFECT DISPLAY PROCESS

S270: EXTRACTS SQUARE AT WHICH ANSWER ROUTE INITIALLY DEVIATES FROM RIGHT ANSWER ROUTE AS FAILURE SQUARE

S272: DISPLAYS MOVING OBJECT ON STARTING POINT END OF ROUTE INDICATOR

S274: CHANGES DISPLAY STATE OF STARTING POINT SQUARE

S276: MOVES MOVING OBJECT ALONG ROUTE INDICATOR, AND DELETES AREA OF ROUTE INDICATOR THROUGH WHICH MOVING OBJECT HAS PASSED

S278: HAS MOVING OBJECT MOVED TO ANOTHER SQUARE?

YES: CHANGES DISPLAY STATE OF SQUARE

NO: RETURN

S280: HAS MOVING OBJECT REACHED FAILURE SQUARE?

YES: CAUSES MOVING OBJECT TO DISAPPEAR

NO: DISPLAYS FAILURE NOTIFICATION

S282: RETURN
FIG. 28

<table>
<thead>
<tr>
<th>PROCESSING SECTION</th>
<th>STORAGE SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAME CALCULATION SECTION</td>
<td>SYSTEM PROGRAM</td>
</tr>
<tr>
<td>E-MAIL CONTROL SECTION</td>
<td>GAME PROGRAM</td>
</tr>
<tr>
<td>DECODER SECTION</td>
<td>MAIL DATA DECODER PROGRAM</td>
</tr>
<tr>
<td>SOUND GENERATION SECTION</td>
<td>E-MAIL PROGRAM</td>
</tr>
<tr>
<td>IMAGE GENERATION SECTION</td>
<td>MAIL DATA</td>
</tr>
<tr>
<td>COMMUNICATION CONTROL SECTION</td>
<td>GAME SCREEN SETTING DATA</td>
</tr>
<tr>
<td>OPERATOIN INPUT SECTION</td>
<td>MOVING OBJECT DATA</td>
</tr>
<tr>
<td>TOUCH POSITION DETECTION SECTION</td>
<td>SQUARE DISPLAY DATA</td>
</tr>
<tr>
<td>SOUND OUTPUT SECTION</td>
<td>PRIZE IMAGE LIBRARY</td>
</tr>
<tr>
<td>IMAGE DISPLAY SECTION</td>
<td>QUESTION CONDITION SETTING DATA</td>
</tr>
<tr>
<td>COMMUNICATION SECTION</td>
<td>RIGHT CHARACTER STRING SET</td>
</tr>
<tr>
<td>RIGHT CHARACTER STRING</td>
<td>ARRAY MODE DATA</td>
</tr>
<tr>
<td>RIGHT CHARACTER STRING ID</td>
<td>WORD DICTIONARY DATA</td>
</tr>
<tr>
<td>RIGHT CHARACTER STRING</td>
<td>DIFFICULTY LEVEL</td>
</tr>
<tr>
<td>QUESTION ARRAY DATA</td>
<td>520</td>
</tr>
<tr>
<td>HINT DISCLOSURE COUNT</td>
<td>524</td>
</tr>
<tr>
<td>ANSWER ROUTE INFORMATION</td>
<td>526</td>
</tr>
<tr>
<td>PLAY TIME</td>
<td>528</td>
</tr>
<tr>
<td>STARTING POINT INDICATOR DATA</td>
<td>530</td>
</tr>
<tr>
<td>ROUTE INDICATOR DATA</td>
<td>532</td>
</tr>
<tr>
<td>SUCCESSIVE RIGHT ANSWER COUNT</td>
<td>534</td>
</tr>
<tr>
<td>ANSWER CHARACTER STRING</td>
<td>536</td>
</tr>
<tr>
<td>CURRENTLY ACQUIRED POINTS</td>
<td>538</td>
</tr>
<tr>
<td>TOTAL ACQUIRED POINTS</td>
<td>540</td>
</tr>
</tbody>
</table>

Oct. 6, 2011 Sheet 25 of 32
FIG. 29

S320 RIGHT CHARACTER STRING SELECTION PROCESS
PERFORMS LOTTERY PROCESS THAT DETERMINES WHETHER OR NOT TO EXTRACT RIGHT CHARACTER STRING FROM EXTERNAL DATA

S322 IS RIGHT CHARACTER STRING EXTRACTION FROM EXTERNAL DATA?
NO

S324 ACQUIRES NUMBER-OF-CHARACTER CONDITION CORRESPONDING TO CURRENT DIFFICULTY LEVEL

S326 DECODES THE MAIL DATA TO OBTAIN TEXT FILE

S328 EXTRACTS SINGLE PHRASE OR RANGE OF PHRASES THAT SATISFIES NUMBER-OF-CHARACTER CONDITION FROM TEXT FILE

S330 SELECTS ONE EXTRACTED PHRASE AND RANGE OF PHRASES, AND USES SELECTED PHRASE AS RIGHT CHARACTER STRING

S332 RIGHT CHARACTER STRING ID= NULL

S340 SELECTS RIGHT CHARACTER STRING SET CORRESPONDING TO CURRENT DIFFICULTY LEVEL

S342 SELECTS ONE RIGHT CHARACTER STRING FROM SELECTED RIGHT CHARACTER STRING SET

S344 STORES SELECTED RIGHT CHARACTER STRING ID AS RIGHT CHARACTER STRING ID

S346 GENERATES RIGHT CHARACTER STRING

END
FIG. 31

QUESTION ARRAY GENERATION PROCESS

S30: SELECTS SETTING OF EACH ITEM OF QUESTION CONDITION BASED ON CURRENT DIFFICULTY LEVEL.

S32: IS QUESTION CONDITION "UNNECESSARY CHARACTER" SET TO OTHER THAN "0"?

YES → 8

NO → S33

S33: ADJACENTLY DISPOSES SQUARES IN SAME NUMBER AS NUMBER OF CHARACTERS OF RIGHT CHARACTER STRING IN RANDOM DIRECTION TO AUTOMATICALLY GENERATE ARRAY MODEL.

S37: ADJACENTLY DISPOSES SQUARES IN SAME NUMBER AS TOTAL OF NUMBER OF CHARACTERS OF RIGHT CHARACTER STRING AND VALUE SET AS QUESTION CONDITION "UNNECESSARY CHARACTER" IN RANDOM DIRECTION TO AUTOMATICALLY GENERATE ARRAY MODEL.

S41: DOES GENERATED ARRAY MODEL SATISFY EACH OF "NUMBER OF CORNERS", "NUMBER OF BENDS", "MAXIMUM NUMBER OF LINEARLY ARRANGED SQUARES", AND "NUMBER OF LEFTWARD STRAIGHT LINES"?

NO → S43

DISCARDS ARRAY MODEL.

YES → 7

S43 → 8
FIG. 32

S70
IS NATURAL NUMBER SET AS QUESTION CONDITION "BLIND BOX"?

S72
HIDES CHARACTERS IN NUMBER CORRESPONDING TO NATURAL NUMBER SET AS QUESTION CONDITION "BLIND BOX"

S82
DOES RIGHT CHARACTER STRING CONSISTS OF ALPHABETICAL CHARACTERS?

S84
IS QUESTION CONDITION "FIRST CHARACTER CAPITALIZATION" SET TO "ON"?

S86
CAPITALIZES FIRST CHARACTER OF RIGHT CHARACTER STRING, AND UNCAPITALIZES REMAINING CHARACTERS

S89
GENERATES QUESTION ARRAY DATA BY BLENDING CHARACTER OF RIGHT CHARACTER STRING WITH SQUARE OF CURRENTLY SELECTED ARRAY MODEL BASED ON ARRANGEMENT ORDER

RETURN
FIG. 34

(1) RIGHT CHARACTER STRING "america"

(2) RIGHT CHARACTER STRING "AMERICA"
FIG. 35

RIGHT CHARACTER STRING “hippopotamus”

(1)

RIGHT ANSWER ROUTE

(2)

RIGHT CHARACTER STRING “hippopotamus”

RIGHT ANSWER ROUTE
GAME EXECUTION METHOD, STORAGE MEDIUM, AND ELECTRONIC INSTRUMENT


BACKGROUND

[0002] A game that sets a question in which a plurality of white pieces and a plurality of black pieces are disposed has been known (e.g., “Chokkan Hitofude” for Nintendo DS (manufactured by Nintendo Co., Ltd.)). The player sequentially touches the pieces with one stroke so that all of the pieces in one row have an identical color. The player intuitively makes a stroke while expecting that the black piece is changed to a white piece, and vice versa.

SUMMARY

[0003] According to one aspect of the invention, there is provided a game execution method comprising:
[0004] displaying a question array generated by adjacently disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
[0005] inputting an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
[0006] comparing an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

[0007] According to another aspect of the invention, there is provided an electronic instrument comprising:
[0008] a question array display control section that displays a question array generated by adjacently disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
[0009] an answer route input section that inputs an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
[0010] a right/wrong determination section that compares an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is an external view illustrative of a configuration example of a portable game device.
[0012] FIG. 2 is a view showing an example of a game screen before the game starts.
[0013] FIG. 3 is a view showing an example of a game screen during an answer operation.
[0014] FIG. 4 is a view showing an example of an effect image displayed when the answer is correct.
[0015] FIG. 5 is a view showing an example of an effect image displayed when the answer is incorrect.
[0016] FIG. 6 is a functional block diagram showing an example of a functional configuration according to a first embodiment.
[0017] FIG. 7 is a view showing an example of the data configuration of a prize image library.
[0018] FIG. 8 is a view showing an example of the data configuration of question condition setting data.
[0019] FIG. 9 is a view showing a setting example of a question condition “number of corners” corresponding to the difficulty level, and an example of a method of counting the number of corners.
[0020] FIG. 10 is a view showing a setting example of a question condition “maximum number of linearly arranged squares” corresponding to the difficulty level, and an example of a method of counting the number of linearly arranged squares.
[0021] FIG. 11 is a view showing a setting example of a question condition “number of bends” corresponding to the difficulty level, and an example of a method of counting the number of linearly arranged squares.
[0022] FIG. 12 is a view showing a setting example of a question condition “number of leftward straight lines” corresponding to the difficulty levels, and an example of a method of counting the number of leftward straight lines.
[0023] FIG. 13 is a view showing a setting example of a question condition “blind box”.
[0024] FIG. 14 is a view showing a setting example of a question condition “first character capitalization”.
[0025] FIG. 15 is a view showing a setting example of a question condition “unnecessary character”.
[0026] FIG. 16 is a view showing a setting example of a question condition “color change”.
[0027] FIG. 17 is a view showing an example of the data configuration of a right character string set.
[0028] FIG. 18 is a view showing an example of the data configuration of an array model data.
[0029] FIG. 19 is a view showing an example of the data configuration of question array data.
[0030] FIG. 20 is a view showing an example of the data configuration of answer route information.
[0031] FIG. 21 is a flowchart illustrative of the flow of a main process according to the first embodiment.
[0032] FIG. 22 is a flowchart that follows the flowchart shown in FIG. 21.
[0033] FIG. 23 is a flowchart illustrative of the flow of a question array generation process according to the first embodiment.
[0034] FIG. 24 is a flowchart that follows the flowchart shown in FIG. 23.
[0035] FIG. 25 is a flowchart illustrative of the flow of an answer route display process.
[0036] FIG. 26 is a flowchart illustrative of the flow of a clear effect display process.
[0037] FIG. 27 is a flowchart illustrative of the flow of a failure effect display process.
[0038] FIG. 28 is a functional block diagram showing an example of a functional configuration according to a second embodiment.
[0039] FIG. 29 is a flowchart illustrative of the flow of a right character string selection process.
FIG. 30 is an external view showing a modification of a game device.

FIG. 31 is a flowchart illustrative of the flow of a modification of a question array generation process.

FIG. 32 is a flowchart that follows the flowchart shown in FIG. 31.

FIG. 33 is a view showing a modification of a clear effect image.

FIG. 34 is a view showing a modification of the configuration of a question array model.

FIG. 35 is a view showing a modification of the configuration of a question array model.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

Several aspects of the invention may make it possible to control execution of a novel puzzle game that utilizes a one-stroke input.

According to one embodiment of the invention, there is provided a game execution method comprising:

- displaying a question array generated by adjacent disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
- inputting an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
- comparing an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

According to another embodiment of the invention, there is provided an electronic instrument comprising:

- a question array display control section that displays a question array generated by adjacent disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
- an answer route input section that inputs an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
- a right/wrong determination section that compares an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

This makes it possible to implement a game that provides characters or the like that form a word or a sentence as the right character string, adjacentely disposse squares to which the characters or the like of the right character string are respectively assigned, and allows the player to input an answer route that reproduces the right character string with one stroke.

The method may further comprise: discriminately displaying squares among the plurality of squares along the right route.

This makes it possible for the player to easily determine the answer route.

The method may further comprise:

- displaying a moving object so that the moving object traces squares among the plurality of squares along the answer route.
- this makes it possible to effectively display whether or not the answer character string is correct.

- The method may further comprise: changing a display state of each square among the plurality of squares through which the moving object moves.
- this makes it possible to clearly indicate the moving path of the moving object, and change the screen in various ways.

The method may further comprise:

- displaying a new image hidden by the question array when it has been determined that the answer character string is correct, and the moving object has reached a final square along the answer route.
- specifically, since a new image is displayed as a result of a right answer, an image can be effectively displayed when the answer character string is correct.

The method may further comprise:

- displaying a hint based on the right character string.
- this makes it possible to present a hint for solving the current question (puzzle).

The method may further comprise:

- displaying a hint by displaying the meaning of the right character string.
- this makes it possible to present the meaning of the right character string as a hint.

The method may further comprise:

- displaying a hint by displaying a first assignment character and/or a final assignment character of the right character string.
- this makes it possible to present the first assignment character and/or the final assignment character of the right character string as a hint.

The method may further comprise:

- displaying a hint by displaying a word included in the right character string.
- this makes it possible to present a word included in the right character string as a hint.

The method may further comprise:

- performing a given effect process based on an elapsed time from a given start timing.
- this makes it possible to improve the game playability by performing the effect process based on the elapsed time from the game start timing.

The method may further comprise:

- performing a filter effect display process that gradually decreases the visibility of each of the plurality of squares based on an elapsed time from a given start timing.
- this makes it possible to gradually increase the difficulty level of the game based on the elapsed time from the game start timing.

The method may further comprise:

- calculating points based on at least an elapsed time from a given start timing.
- this makes it possible to calculate points based on the elapsed time until an answer is input.

The method may further comprise:

- calculating points based on whether or not the hint has been displayed.
This makes it possible to calculate points based on whether or not the hint has been displayed.

The method may further comprise:

determining the right character string by reading characters, numerals, signs, or a combination thereof having a meaning from a private database that is selected from a directory, a telephone book, a diary, a schedule book, and e-mail data and can be edited by a player by performing an operation input.

Specifically, since the right character string can be determined from the player’s private information, a more intimate game can be implemented.

According to another embodiment of the invention, there is provided a non-transitory storage medium storing a program that causes a computer to execute the method according to claim 1.

The term “storage medium” used herein refers to a magnetic disk, an optical disk, an IC memory, and the like.

First Embodiment

A first embodiment to which the invention is applied illustrates an example in which a portable game device executes a one-stroke puzzle game that dispenses characters (e.g., character, numeral, or sign) having a meaning respectively in a plurality of adjacent disposed squares, and allows the player to input an answer route by tracing all or some of the squares along a one-way route to find the original character string. Note that the term “character” used herein refers to characters of various languages such as a hiragana character, a katakana character, a Chinese character, an alphabetical character, and a Hangul character, a numeral, a sign, and the like, unless otherwise indicated. The term “assignment character” refers to at least one character, numeral, sign, or a combination thereof that is assigned to one square. The following embodiments are mainly described taking an example in which one character is assigned to one square.

Configuration of Game Device

FIG. 1 is an external view illustrative of a configuration example of a portable game device. A portable game device 1400 according to this embodiment includes an arrow key 1402 and button switches 1404 that allow the player to input a game operation, a first liquid crystal display 1406, a second liquid crystal display 1408, a speaker 1410, a control unit 1450, and a flip-top main body 1401 that can be opened and closed through a hinge 1414, the arrow key 1402, the button switches 1404, the first liquid crystal display 1406, the second liquid crystal display 1408, the speaker 1410, and the control unit 1450 being provided on or in the main body 1401. Touch panels 1407 and 1409 are provided on the surface of the first liquid crystal display 1406 and the surface of the second liquid crystal display 1408, respectively, the touch panels 1407 and 1409 allowing the player to designate an arbitrary position within the display screen by performing a touch operation using a stylus pen 1416 or the like.

The main body 1401 includes a memory card reader 1418 that reads and writes data from and into a memory card 1440 (i.e., computer-readable information storage medium). The memory card 1440 stores a program and setting data necessary for the control unit 1450 of the portable game device 1400 to perform various game calculation processes. The main body 1401 is also provided with a built-in battery, a power button, a volume control button, and the like (not shown).

The touch panels 1407 and 1409 respectively cover almost the entire display screen of the first liquid crystal display 1406 and the second liquid crystal display 1408 so that images displayed on the first liquid crystal display 1406 and the second liquid crystal display 1408 can be observed from the outside. When the player has touched the touch panel 1407 or 1409 using the stylus pen 1416 (or a finger or the like) as if to touch the displayed object, the touch position coordinates in an orthogonal coordinate system (origin: upper left) can be output to the control unit 1450.

The control unit 1450 corresponds to a control board of a game device, and includes a processor (e.g., central processing unit (CPU), graphics processing unit (GPU), and digital signal processor (DSP)), an application-specific integrated circuit (ASIC), and an IC memory (e.g., VRAM, RAM, and ROM).

The control unit 1450 also includes a wireless communication module 1412, a three-axis acceleration sensor 1422, a driver circuit that drives the first liquid crystal display 1406 and the liquid crystal display 1408, a driver circuit that drives the touch panel 1407 and the touch panel 1409, a circuit that receives signals from the arrow key 1402 and the button switches 1404, and an amplifier circuit that outputs a sound signal to the speaker 1124, and an interface circuit (I/F circuit) such as a signal input-output circuit that exchanges signals with the memory card reader 1418. The elements included in the control unit 1450 are electrically connected via a bus circuit so that data can be read from and written into each element, or a signal can be exchanged between each element.

The three-axis acceleration sensor 1422 detects accelerations in an X-axis direction, a Y-axis direction, and a Z-axis direction that perpendicularly intersect to detect a change in posture or position of the portable game device 1400, and outputs a detection signal to the control unit 1450. Note that the portable game device 1400 may include a gyroscope sensor instead of, or in addition to, the acceleration sensor. When detecting a change in position or posture of the portable game device 1400 based on terrestrial magnetism, the acceleration sensor may be replaced with a magnetic sensor.

The control unit 1450 reads a program and data stored in the memory card 1440 via the memory card reader 1418, and temporarily stores the program and data in the IC memory. The control unit 1450 performs a calculation process by executing the program read from the memory card 1440, and controls each section of the portable game device 1400 (executes an action game) based on operation inputs from the arrow key 1402, the button switches 1404, and the touch panels 1407 and 1409.

Although this embodiment employs a configuration in which the portable game device 1400 reads a necessary program and setting data from the memory card 1440, it is also possible to employ a configuration in which the portable game device 1400 connects to a cable/wireless communication channel 1 (e.g., Internet, local area network (LAN), or wide area network (WAN)) through the wireless communication module 1412, and downloads a necessary program and setting data from an external device.

Description of Game

The details of the puzzle game according to this embodiment are described below. A game screen W2 shown in FIG. 2 is displayed when the game has started, for example. The first liquid crystal display 1406 displays a hint display area 2 (2a, 2b, 2c), a level 4 of the current question, acquired
points 6, and a time limit 8 that indicates the remaining play time. The second liquid crystal display 1408 displays a message display area 10, a main operation area 12, a hint button 14 that allows the player to input a hint disclosure operation by performing a touch operation, and an answer completion button 16 that allows the player to input an answer completion operation.

[0109] The details of a hint are not displayed in the hint display area 2 when the game has started (see the hint display areas 2a and 2c shown in FIG. 2). When the player has performed a given hint disclosure operation, the details of a hint are displayed in the hint display area 2 (see the hint display area 2a).

[0110] A game operation guide image, a game status notification image, a cheer message, and the like are displayed in the message display area 10.

[0111] A question array model 20 is displayed in the main operation area 12. The question array model 20 is an array model formed by adjacent disposing a plurality of polygons (quadrilaterals in the example shown in FIG. 2) (hereinafter referred to as “squares”), an assignment character being drawn in each square. The term “assignment character” used herein refers to main characters of each language (e.g., hiragana character, katakana character, Chinese character, and alphabetical character), numerals, signs, units, and the like. The assignment character may include one character, or may include two or more characters. Note that the term “assignment character” is hereinafter referred to as “character string”.

[0112] A character string is assigned so that a specific meaning is obtained by reading the character string along a one-way route from one square to an adjacent square so that each square is passed only once. The question array model 20 is fanned so that the player cannot easily determine the route along which the player should read the character string at first sight. The question array model 20 is formed so that the player cannot easily determine whether the character string is completed using all or some of the characters displayed in the question array model 20 at first sight.

[0113] FIG. 3 is a schematic view showing an example of a state from the start of the game to completion of an answer.

[0114] As shown in FIG. 3 (see (1), game screen W4), the player determines the starting point by touching one square of the question array model 20 in which the first character of the character string is considered to be displayed. A starting point indicator 30 is displayed corresponding to the square touched by the player.

[0115] The player then moves the touch position to one square in which the second character of the character string is considered to be displayed among the squares adjacent to the square determined to be the starting point (starting point square) without removing the stylus pen 1416 from the touch panel. When the player has removed the stylus pen 1416 from the touch panel after touching the starting point square, the player may touch (select) the starting point square again.

[0116] When the player has selected one square in which the second character of the character string is considered to be displayed, the player then moves the touch position to one square in which the third character of the character string is considered to be displayed without removing the stylus pen 1416 from the touch panel. The above operation is repeated until the character string is completed.

[0117] As shown in FIG. 3 (see (2)), a route indicator 32 sequentially extends from the starting point square so that the route indicator 32 connects the squares sequentially touched by the player. When the player has determined that the player has reached the square in which the final character of the character string is considered to be displayed, the player touches the answer completion button 16, and waits for the game device to determine whether or not the answer is right.

[0118] When the player has performed the answer completion operation, an answer character string is generated by arranging the characters drawn in the squares along the input route. The answer character string is then compared with the question character string (right character string) to determine whether or not the answer is right. An effect image is then displayed based on whether or not the answer is right.

[0119] FIG. 4 is a view showing an example of a clear effect image according to this embodiment that is displayed when the answer is right. As shown in FIG. 4 (see (1)), game screen W10, a flame (moving object 34) appears from the starting point end of the route indicator 32, and gradually moves toward the end point of the route indicator 32 as if the route indicator 32 were a fuse. The square through which the moving object 34 has passed is changed in display state (i.e., burned) (see the upper left square in (1)).

[0120] When the moving object 34 has reached the end point of the route indicator 32, the moving object 34 disappears, and all of the squares have been burned, as shown in FIG. 4 (see (2)), game screen W12. These squares are then broken down, and a prize image 36 is displayed as a new image, as shown in FIG. 4 (see (3)), game screen W14. The prize image 36 indicates that the current answer is right.

[0121] An advance operation button 38 that allows the player to continue the game by touching the advance operation button 38, and a game finish button 40 that allows the player to finish the game by touching the game finish button 40 are displayed within the game screen W14. The player arbitrarily touches the advance operation button 38 or the game finish button 40. When the player has touched the advance operation button 38, a new question character string is selected, and the game screen W2 is displayed. When the player has touched the game finish button 40, the game ends.

[0122] FIG. 5 is a view showing an example of a failure effect image according to this embodiment that is displayed when the answer is wrong. As shown in FIG. 5 (see (1), game screen W16), a flame (moving object 34) appears from the starting point end of the route indicator 32, and gradually moves toward the end point of the route indicator 32 as if the route indicator 32 were a fuse in the same manner as the game screen W10.

[0123] As shown in FIG. 5 (see (2), game screen W18), the moving object 34 disappears at a point where the answer route extends to a wrong square as if the flame went out. In the example shown in FIG. 5 (see (2)), the answer route is right from the starting point to the second square on the right, but is wrong when the answer route extends to the bottom square on the right (i.e., first wrong point). When the moving object 34 has disappeared, a message that states that the answer route is wrong is displayed in the message display area 10, as shown in FIG. 5 (see (3)), game screen W20.

[0124] A retry button 42 that allows the player to retry the same question by touching the retry button 42, and a game finish button 40 are displayed within the game screen W20. When the player has touched the retry button 42, the game screen W2 is displayed, and the same question character

string is displayed. When the player has touched the game finish button 40, the game ends.

[0125] Functional Configuration

[0126] FIG. 6 is a functional block diagram showing an example of the functional configuration according to this embodiment that implements the above game.

[0127] The game device 1400 according to this embodiment includes an operation input section 100, a processing section 200, a sound output section 350, an image display section 360, a communication section 370, and a storage section 500.

[0128] The operation input section 100 outputs an operation input signal to the processing section 200 based on an operation input performed by the player. The operation input section 100 may be implemented by a button switch, a joystick, a touch pad, a trackball, a multi-axis acceleration sensor that has two or more detection axes, a single-axis acceleration sensor unit formed by combining acceleration sensors so that the detection axis direction differs, a multi-direction tilt sensor that has two or more detection directions, a single-direction tilt sensor unit formed by combining tilt sensors so that the detection direction differs, and the like. The arrow key 1402 and the button switches 1404 shown in FIG. 1 correspond to the operation input section 100.

[0129] The operation input section 100 includes a touch position detection section 102.

[0130] The touch position detection section 102 is implemented by a device that detects a touch position within a display screen. The touch panels 1407 and 1409 shown in FIG. 1 correspond to the touch position detection section 102.

[0131] The processing section 200 is implemented by a microprocessor (e.g., CPU and GPU), an application-specific integrated circuit (ASIC), an IC memory, and the like. The processing section 200 exchanges data with (controls data exchange between) each functional section of the portable game device 1400 including the operation input section 100 and the storage section 500. The processing section 200 controls the operation of the portable game device 1400 by performing various calculation processes based on a given program, data, and the operation input signal from the operation input section 100. In FIG. 1, the control unit 1450 corresponds to the processing section 200.

[0132] The processing section 200 according to this embodiment includes a game calculation section 210, a sound generation section 250, an image generation section 260, and a communication control section 270.

[0133] The game calculation section 210 performs various processes for implementing the puzzle game according to this embodiment. Specifically, the game calculation section 210 includes a right character string determination section 212, a difficulty level setting section 214, a question array generation section 216, a hint display control section 218, an answer route management section 220, an answer character string generation section 222, a right/wrong determination section 224, an effect control section 226, a point calculation section 228, and a filter effect control section 229.

[0134] The right character string determination section 212 determines a question character string (right character string). Specifically, the right character string determination section 212 selects a right character string set 514 corresponding to the difficulty level determined based on the game results of the player from a plurality of right character string sets 514 stored in the storage section 500, and selects one of the character strings registered in the selected right character string set 514. Identification information about the selected character string is stored in the storage section 500 as a right character string ID 522, and the selected character string is stored in the storage section 500 as a right character string 524.

[0135] The difficulty level setting section 214 determines a difficulty level 520 of the question based on the game results of the player, and stores the difficulty level 520 in the storage section 500. In this embodiment, the difficulty level 520 is initially set to “1”, and gradually increased to “5” based on the successive right answer count (successive right answer count 544) (i.e., the number of times that the player has successively answered correctly). Note that the difficulty level 520 may be appropriately set. The condition whereby the difficulty level 520 is increased is not limited to the successive right answer count, but may be appropriately set.

[0136] The question array generation section 216 generates the question array model 20 based on the difficulty level 520 (see FIG. 2).

[0137] Specifically, the question array generation section 216 selects a condition (question condition) that defines features that should be included in the question based on the difficulty level 520 from question condition setting data 512 stored in the storage section 500. The question array generation section 216 then selects an array model (i.e., a model formed by adjacent disposition of identical polygons (squares in which a character is not drawn)) that satisfies the selected question condition from an array model data 516 stored in the storage section 500.

[0138] The array model data 516 is provided corresponding to the number of squares, and the array model is linked to the arrangement order that indicates the order in which the right character string 524 is disposed (described later). An image of each character of the right character string 524 is blended (synthesized) with each square based on the arrangement order to generate the question array model 20. Information about the generated question array model 20 is stored in the storage section 500 as question array data 526 together with information about the character assigned to each square.

[0139] The hint display control section 218 displays a hint that relates to the current right character string 524 when the player has performed a given hint disclosure operation. Specifically, the hint display control section 218 displays a hint that is linked in advance to the right character string 524 in the right character string set 514 (see the hint display area 2 shown in FIG. 2).

[0140] The answer route management section 220 performs a display control process that displays the route indicator 32 based on the square touch operation performed by the player so that the squares positioned along the answer route are displayed, stores the order and the route of the squares of the question array model 20 (see FIG. 2) touched by the player in the storage section 500 as answer route information 532, and manages the answer route information 532.

[0141] The answer character string generation section 222 generates an answer character string 546 by arranging the characters assigned to the squares touched by the player based on the answer route stored as the answer route information 532 when the player has performed a given answer completion operation, and stores the answer character string 546 in the storage section 500.

[0142] The right/wrong determination section 224 compares the right character string 524 with the answer character string 546 to determine whether or not the answer is right.
The effect control section 226 displays a clear effect or a failure effect based on the determination result of the right/wrong determination section 224 (see FIGS. 4 and 5).

The point calculation section 228 calculates points using the time elapsed before the player has performed the answer completion operation after the display start timing of the question array model 20 (i.e., given start timing).

The filter effect control section 229 gradually decreases the visibility of the squares based on the elapsed time after the game has started. Specifically, the filter effect control section 229 blends an image that decreases in transparency (i.e., becomes opaque) with the lapse of time with part or the entirety of the question array model 20. The visibility of the squares may be decreased by changing the color of the character or the like drawn in the square to be closer to the color of the square with the lapse of time, gradually or quickly rotating the character or the like drawn in the square with the lapse of time, or mosaicing the square with the lapse of time, for example.

The sound generation section 250 is implemented by a processor (e.g., digital signal processor (DSP) or sound synthesis IC) and an audio codec that can reproduce a sound file, for example. The sound generation section 250 generates a sound signal of an effect sound, background music (BGM), or an operation sound based on the processing results of the game calculation section 210, and outputs the generated sound signal to the sound output section 350.

The sound output section 350 is implemented by a device that outputs sound such as an effect sound or BGM based on the sound signal input from the sound generation section 250. The speaker 1410 shown in FIG. 1 corresponds to the sound output section 350.

The image generation section 260 is implemented by a processor (e.g., graphics processing unit (GPU) or a digital signal processor (DSP)), a video signal IC, a program (e.g., video codec), a drawing frame memory (e.g., frame buffer), and the like. The image generation section 260 generates a game screen (image) every frame (e.g., 1/60th of a second) based on the processing results of the game calculation section 210, and outputs an image signal of the generated game screen (image) to the image display section 360.

The image display section 360 displays a game image based on the image signal input from the image generation section 260, which is implemented by an image display device such as a flat panel display, a cathode-ray tube (CRT), a projector, or a head mount display. The first liquid crystal display 1406 and the second liquid crystal display 1408 shown in FIG. 1 correspond to the image display section 360.

The communication control section 270 performs a data communication process, and exchanges data with an external device via the communication channel 1, and implements communication. The communication section 370 connects to the communication channel 1, and implements communication. The communication section 370 is implemented by a transceiver, a modem, a terminal adapter (TA), a jack for a communication cable, a control circuit, and the like. The wireless communication module 1412 shown in FIG. 1 corresponds to the communication section 370.

The storage section 500 stores a system program 501 and a game program 502. The system program 501 implements the basic input/output function of the portable game device 1400 as a computer. The game program 502 is application software that is read and executed by the processing section 200 so that the processing section 200 implements the functions of the game calculation section 210. The game program 502 may be incorporated in the system program 501.

The storage section 500 stores the game setting data 504, the moving object data 506, the square display data 508, the prize image library 510, the question condition setting data 512, a plurality of right character string sets 514, a plurality of pieces of array model data 516, a dictionary data 518, the starting point indicator data 540, and the route indicator data 542. These data are provided in advance.

The storage section 500 also stores a difficulty level 520, a right character string ID 522, a right character string 524, a question array data 526, a hint disclosure count 530, an answer route information 532, a play time 534, a successive right answer count 544, an answer character string 546, currently acquired points 550, and total acquired points 552 as information that is generated and stored during the game. The storage section 500 also appropriately stores data (e.g., timer value and counter value) that is required for controlling the game process.

The game screen setting data 504 includes information that is used to display the hint display area 2, the level display 4, the acquired points 6, the time limit 8, the message display area 10, the hint button 14, the answer completion button 16, the advance operation button 38, the game finish button 40, the retry button 42, and the like at given positions within the game screens W2 to W20 (see FIGS. 2 to 8).

The moving object data 506 includes image data for displaying the moving object 34 (see FIG. 4), and the like.

The prize image library 510 includes image data about the prize image that appears at the end of the clear effect display image. As shown in FIG. 7, the prize image library 510 includes a currently acquired point condition 510a and prize image data 510b, for example. Note that the prize image data 510b may not be linked to the acquired points. For example, the prize image data 510b may be linked to the play time or the number of questions. When linking the prize image data 510b to the number of questions, it is preferable to provide consecutive images as the prize image data 510b so that the story advances, or the time elapses as the number of questions increases. For example, the prize image data 510b may be provided so that the number of flowering cherry blossom trees increases as the number of questions increases.

The question condition setting data 512 includes a condition (question condition) that defines features that should be included in the question array model 20 based on the difficulty level 520. As shown in FIG. 8, the question condition setting data 512 includes an applied level setting...
items of the question condition, and the details thereof, for example. In this embodiment, a plurality of question conditions can be set corresponding to each difficulty level.

The items of the question condition are described in detail below.

FIG. 9 is a view showing a setting example of a question condition “number of corners” corresponding to the difficulty level, and an example of a method of counting the number of corners. The question condition “number of corners” refers to the number of squares of the question array model 20 that correspond to a “corner.” In this embodiment, the number of corners permitted at each difficulty level is designated. When the square is quadrilateral, a square for which the number of adjacent squares is 2 or less is determined to be a corner.

Since the number of squares adjacent to the corner is small compared with a square other than the corner, the player can more easily determine the right answer route as the number of corners increases, and vice versa. For example, the square that protrudes upward in FIG. 9 (see (3)) must be the starting point or the end point of the answer route. Therefore, the player can easily determine the right answer route as compared with an example (2) shown in FIG. 9. Accordingly, the question condition is set so that the number of corners increases as the difficulty level decreases, and decreases as the difficulty level increases (see (1) in FIG. 9).

FIG. 10 is a view showing a setting example of a question condition “maximum number of linearly arranged squares” corresponding to the difficulty level, and an example of a method of counting the number of linearly arranged squares. The question condition “maximum number of linearly arranged squares” refers to the maximum number of squares that are linearly arranged along the right arrangement route (order). In this embodiment, the maximum length permitted at each difficulty level is designated. The squares may be linearly arranged in an arbitrary direction.

For example, the number of squares and the external shape of the array model are identical in the examples (2) and (3) shown in FIG. 10. When the maximum number of linearly arranged squares indicated by a broken arrow is “5” (see (2) in FIG. 10), since the characters stored as the right character string 524 are linearly arranged, the player can easily determine the right answer route. When the maximum number of linearly arranged squares is “3” (see (3) in FIG. 10), since the characters stored as the right character string 524 are linearly arranged within a small number of squares, the player cannot easily determine the right answer route. Accordingly, the question condition is set so that the maximum number of linearly arranged squares increases as the difficulty level decreases, and decreases as the difficulty level increases (see (1) in FIG. 10).

Note that the average value or the median value of the numbers of linearly arranged squares may also be used.

FIG. 11 is a view showing a setting example of a question condition “number of bends” corresponding to the difficulty level, and an example of a method of counting the number of linearly arranged squares. The question condition “number of bends” refers to the total number of bends of the right route. In this embodiment, the number of bends permitted at each difficulty level is designated.

For example, the number of squares and the external shape of the array model are identical in the examples (2) and (3) shown in FIG. 11. However, the number of bends in the example (2) shown in FIG. 11 is smaller than that of the example (3) shown in FIG. 11. When the number of bends is small, the player can easily determine the right answer route since the number of linearly arranged squares increases along the right answer route. Accordingly, the question condition “number of bends” is set so that the question condition “number of bends” decreases as the difficulty level decreases, and increases as the difficulty level increases (see (1) in FIG. 11).

Since the question conditions “maximum number of linearly arranged squares” and “number of bends” are parameters having a correlation, one of the question conditions “maximum number of linearly arranged squares” and “number of bends” may be omitted.

FIG. 12 is a view showing a setting example of a question condition “number of leftward straight lines” corresponding to the difficulty level, and an example of a method of counting the number of linearly arranged squares. The question condition “number of leftward straight lines” refers to the number of areas in which two or more squares are linearly arranged in the leftward direction along the right route. The number of leftward straight lines permitted at each difficulty level is designated.

For example, the number of squares and the external shape of the array model are identical in the examples (2) to (4) shown in FIG. 12. However, the number of leftward straight lines indicated by a broken arrow that indicates the right route differs between the examples (2) to (4) shown in FIG. 12. Japanese characters and English characters are normally written in the rightward direction. Therefore, while an area in which characters are arranged in the rightward direction is easily recognized as a character string having a meaning, an area in which characters are arranged in the leftward direction is not easily recognized as a character string having a meaning. Therefore, the player cannot easily determine the right answer route as the number of leftward straight lines increases. Accordingly, the question condition “number of leftward straight lines” is set so that the question condition “number of leftward straight lines” decreases as the difficulty level decreases, and increases as the difficulty level increases (see (1) in FIG. 12).

The above also applies to the number of areas in which two or more squares are linearly arranged in the upward direction along the right route. Therefore, the item “number of leftward straight lines” may be replaced by the item “number of upward straight lines”, or may be used in combination with the item “number of upward straight lines”. The item “number of leftward straight lines” may be replaced by the item “number of rightward straight lines” for a language in which characters are normally written in the leftward direction.

FIG. 13 is a view showing a setting example of a question condition “blind box”. The question condition “blind box” refers to the number of squares in which the character is not displayed. In this embodiment, the blind box permitted at each difficulty level is designated. Specifically, the player cannot easily determine the right answer route as the number of squares in which the character is replaced by a given blind sign or the like increases. Accordingly, the question condition “blind box” is set so that the question condition “blind box” decreases as the difficulty level decreases, and increases as the difficulty level increases.

The “blind sign” may be a question mark or the like. Alternatively, the character may be hidden, or mosaiced, for example. In the latter case, part or the entirety of the character may be hidden.
FIG. 14 is a view showing a setting example of a question condition “first character capitalization”. Note that the capital letters “WIKIPEDIA” of a right character string “wikipedia” shown in FIG. 14 have been registered as a trademark. The question condition “first character capitalization” refers to a setting that indicates whether to capitalize the first character of the character string when the right character string 524 is indicated by small letters (alphabetical characters). When the first character of the character string is capitalized, the player can easily determine the square that corresponds to the starting point. Therefore, the difficulty level decreases. Accordingly, the first character of the character string is capitalized when the difficulty level is relatively low.

A question condition “first character special display” may be set to achieve a similar effect. When the question condition “first character special display” is set to “ON”, the first character is displayed in a special display state (e.g., font type, font size, font color, or blink pattern is changed). The color, pattern, or contour of the corresponding square may be displayed in a special display state instead of (or in addition to) the character.

FIG. 15 is a view showing a setting example of a question condition “unnecessary character”. The question condition “unnecessary character” refers to adding a character irrelevant to the characters of the right character string 524. In this embodiment, the number of unnecessary characters permitted at each difficulty level is designated. In the example shown in FIG. 15, unnecessary characters “v” and “z” are arranged at the front of the right character string “fantastic”. In FIG. 15, an area in which the right character string is arranged is indicated by a broken arrow, and an area in which the unnecessary characters are arranged is indicated by a dash-dotted arrow. Specifically, the player can easily determine the right answer route when unnecessary characters are not added. Accordingly, the question condition “unnecessary character” is set so that the question condition “unnecessary character” decreases as the difficulty level decreases, and increases as the difficulty level increases.

FIG. 16 is a view showing a setting example of a question condition “color change”. The question condition “color change” refers to changing the square display state corresponding to each word of the right character string. Specifically, the player can easily determine the right answer route when the square display state is changed corresponding to each word (see (3) in FIG. 16) as compared with the case where all of the squares are displayed in an identical display state (see (1) in FIG. 16). Accordingly, the question condition “color change” is set to “ON” when the difficulty level is relatively low, and set to “OFF” when the difficulty level is relatively high.

Note that the character display state may be changed instead of the square display state.

A question condition “first square arrangement” designates whether to set the first square along the right route in a peripheral area, a corner area, or a protrusion area (i.e., whether to dispose the first square of the question array model 20 in a peripheral area, a corner area, or a protrusion area of the question array model 20). Since the number of adjacent squares is small in the peripheral area, the corner area, or the protrusion area of the question array model 20 as compared with the center square, the player can easily determine the right answer route if the first square is disposed in the peripheral area, the corner area, or the protrusion area. Accordingly, the question condition “first square arrangement” is set to “ON” when the difficulty level is relatively low, and set to “OFF” when the difficulty level is relatively high.

In this embodiment, the question condition setting data 512 (see FIG. 8) includes all of the above items. Note that the items may be selectively used.

FIG. 17 is a view showing an example of the data configuration of the right character string set 514. The right character string set 514 includes an applied level 514a (i.e., an applied difficulty level that designates the difficulty level to which the right character string set is applied), and a number-of-character condition 514b for the right character string defined by the right character string set 514. The right character string set 514 includes a right character string ID 514d, a right character string 514e, and data that indicates a hint (e.g., a first hint 514f that indicates the meaning of the right character string, and a second hint 514g that indicates the first character and/or the final character of the right character string) (i.e., options that satisfy the number-of-character condition 514b).

Note that the hint may be appropriately set. For example, the hint may be a word included in the right character string. The hint may be an image or sound instead of a text.

FIG. 18 is a view showing an example of the data configuration of the array model data 516. The array model data 516 defines a set of squares to which the character string indicated by the right character string 524 is assigned. The array model data 516 includes an array ID 516a, a square count 516b (i.e., the total number of squares arranged), an array model 516c, array coordinates 516d, and an arrangement order 516e.

The array model 516c indicates a model formed by adjacently disposing a plurality of squares corresponding to the number indicated by the square count 516b. A character is not drawn on the model indicated by the array model 516c.

The array coordinates 516d define coordinates (P1 to P5) of a representative point of each square indicated by the array model 516c based on a local origin P0. The display position of the game screen is adjusted so that the local origin P0 is approximately positioned at the center of the main operation area 12. The arrangement order 516e defines the order in which the character string indicated by the right character string 524 is assigned to each square of the array model 516c. Therefore, a plurality of pieces of array model data 516 may be set based on the arrangement order 516e even if the number of squares and the external shape indicated by the array model 516c are identical.

The word dictionary data 518 is dictionary data used to identify a word from arbitrary text data.

The right character string ID 522 includes a right character string ID 514d used for the current question.

The right character string 524 includes a copy of the right character string 514e corresponding to the right character string ID 514d stored as the right character string ID 522, and is appropriately changed based on the applied question condition. For example, when the question condition “blind box” is set to “1”, one of the characters is replaced by a question mark or the like.

The question array data 526 includes setting information about the question array model 20 displayed in the main operation area 12 within the game screen. As shown in FIG. 19, the question array data 526 includes a question array model 20 and a character link data 526b, for example. The question array model 20 is generated by selecting the array...
model 516c that satisfies the question condition from the array model data 516, and sequentially blending the character string of the right character string 524 with the array model 516c based on the arrangement order 516e. The character link data 526b defines the character assigned to each square based on the arrangement order 516c.

The hint disclosure count 530 is initially set to “0”, and is incremented by one each time the player has performed a given hint disclosure operation.

The answer route information 532 includes (stores) the history of the touch operation performed on the array model 20 displayed in the main operation area 12 within the game screen. As shown in FIG. 20, the answer route information 532 includes a touch order 532d and touched square representative point coordinates 532b, for example.

The play time 534 includes (stores) the elapsed time after the array model 20 has been displayed in the main operation area 12 within the game screen.

The successive right answer count 544 includes (stores) the number of times that the player has successively answered correctly.

The answer character string 546 includes (stores) a character string obtained by arranging the characters linked to the squares of the question array model 20 along the answer route referring to the answer route information 532 when the player has performed a given answer completion operation.

Process Flow

The flow of the process performed by the portable game device 1400 according to this embodiment is described below. The following process is implemented by causing the processing section 200 to read and execute the program 501 and the game program 502.

Generation and output of the image signal of the game screen, and generation and output of the sound signal of the game sound are briefly described below.

Specifically, the image generation section 260 generates and outputs the image signal for displaying the game screen in a cycle sufficiently shorter than the refresh rate of the image display section 360 (first liquid crystal display 1406 and second liquid crystal display 1408). When generating the game screen by DCDG, the image generation section 260 performs a rendering process and the like. The sound generation section 250 generates the sound signal of the game sound, and the sound output section 350 (speaker 1410) outputs the game sound.

FIGS. 21 and 22 are flowcharts illustrative of the flow of the main process according to this embodiment. The processing section 200 initializes the difficulty level 520 to “1”, and initializes the successive right answer count 544 to “0” (step S2). The processing section 200 then selects the right character string set 514 corresponding to the current difficulty level 520 (step S4). The processing section 200 selects one right character string 514e from the selected right character string set 514 (step S6), and stores the right character string ID 514d corresponding to the selected right character string 514e as the right character string ID 522 (step S8).

The processing section 200 copies the character string indicated by the selected right character string 514e, and stores the copied character string as the right character string 524 (step S10).

The processing section 200 then displays each area of the game screen other than the question array model 20 (step S12; see FIG. 2), and performs a question array generation process (step S14).

FIGS. 23 and 24 are flowcharts illustrative of the flow of the question array generation process according to this embodiment. The processing section 200 refers to the question condition setting data 512, and selects the setting of each item of the question condition based on the current difficulty level 520 (step S30). Specifically, the processing section 200 selects the applied level setting 512 corresponding to the current difficulty level 520 from the question condition setting data 512, and selects the question condition setting 512b of each item corresponding to the selected applied level setting 512.

When the question condition “unnecessary character” is set to “0” (NO in step S32), the processing section 200 selects one of the array model data 516 for which the square count 516b is equal to the number of characters of the right character string 524 (step S34; see FIG. 18). When the question condition “unnecessary character” is set to a value other than “0” (YES in step S32), the processing section 200 updates the right character string 524 by adding (inserting) characters in a number corresponding to the question condition “unnecessary character” at the head or the end of the right character string 524 (step S35). The processing section 200 then selects one of the array model data 516 for which the square count 516b is equal to the total number of characters of the right character string 524 that has been updated by adding characters in a number corresponding to the question condition “unnecessary character” (step S36).

The processing section 200 then determines whether or not the array model data 516 selected in the step S34 or S36 satisfies each of the items “number of corners”, “number of bends”, “maximum number of linearly arranged squares”, “number of leftward straight lines”, and “first square arrangement” of the question condition selected in the step S30. When the processing section 200 has determined that the above items are not satisfied (NO in step S40), the processing section 200 discards the current selection of the array model data 516 (step S42), returns to the step S32, and selects another array model data 516.

When the processing section 200 has determined that the above items are satisfied in the step S40 (YES in step S40), the processing section 200 transitions to the subsequent process.

As shown in FIG. 24, the processing section 200 corrects the right character string 524 to match the setting of the question condition “blind box”. Specifically, when a natural number is set as the question condition “blind box” (YES in step S70), the processing section 200 does not display (hides) characters of the character string indicated by the right character string 524 in a number corresponding to the natural number set as the question condition “blind box” (step S72). The characters may be replaced by a question mark, or may be mosaiced. When “0” is set as the question condition “blind box” (NO in step S70), this process is not performed.

The processing section 200 then refers to the setting of the question condition “first character capitalization” (step S84). When the question condition “first character capitalization” is set to “ON” (YES in step S84), the processing section 200 capitalizes the first character of the right character string 524, and uncapitalizes the remaining characters (step S86).

The array model data 516 and the right character string 524 that satisfy the question condition have thus been provided. The processing section 200 then generates the question array data 526 that defines the question array model.
20 used for the current question (step S88). Specifically, the processing section 200 blends an image of each character of the current right character string 524 with the corresponding square of the array model 516c of the currently selected array model data 516 based on the arrangement order 516c to generate the question array model 20. The processing section 200 also generates the character link data 526b, and stores it as the question array data 526. The processing section 200 thus completes the question generation process.

[0207] Again referring to FIG. 21, the processing section 200 then initializes the hint disclosure count 530 to “0” (step S100), clears the answer route information 532 (step S102), and displays the question array model 20 in the main operation area 12 within the game screen (step S104; see FIG. 2). The processing section 200 then measures the play time 534, and starts to display the time limit 8 (step S106). The processing section 200 performs the filter effect display process that decreases the visibility of the array model 20 based on the elapsed time from the play start timing (step S108). For example, the processing section 200 calculates the transparency so that the transparency decreases from 100% with the lapse of time, and blends a given filter image with the array model 20 to achieve the calculated transparency. Alternatively, the number of mosaicd areas having a given size and set to the array model 20 may be increased with the lapse of time. As the effect display based on the elapsed time from the start timing, a moving image (e.g., red rotating image) that causes the player to get tense may be displayed in the main operation area 12 as a background image, for example.

[0208] The processing section 200 determines that the player has performed the hint disclosure operation when it has been detected that the hint button 14 within the game screen has been touched (YES in step S110). When the hint disclosure count 530 has not reached a given upper limit (NO in step S112), the processing section 200 refers to the right character string set 514, reads the nth hint (n is a natural number) corresponding to the current hint disclosure count 530 from the hint information about the right character string ID 514d corresponding to the right character string ID 522, and displays the nth hint in the hint display area 2 (step S114).

[0209] When the processing section 200 has detected that the player has touched the question array model 20 displayed within the game screen (YES in step S120), the processing section 200 performs an answer route display process (step S122).

[0210] FIG. 25 is a flowchart illustrative of the flow of the answer route display process according to this embodiment. Specifically, the processing section 200 determines whether or not the currently touched square is the starting point (step S140). The processing section 200 determines that the touched square is the starting point when no historical information has been stored as the answer route information 532 (YES in step S140), and displays the starting point indicator 30 in the square (step S142; see FIG. 3).

[0211] The processing section 200 then determines whether or not the currently touched square is adjacent to the final square along the answer route (step S143). When the processing section 200 has determined that the currently touched square is not adjacent to the final square along the answer route (NO in step S143), the processing section 200 finishes the answer route display process without performing the process of the steps S144 to S152. When the processing section 200 has determined that the currently touched square is adjacent to the final square along the answer route (YES in step S143), the processing section 200 determines whether or not the currently touched square is a square preceding the final square along the answer route (step S144). When the processing section 200 has determined that the currently touched square is not a square preceding the final square along the answer route based on the answer route information 532, the processing section 200 determines that the player has touched another square to extend the answer route (NO in step S144). The processing section 200 then stores the coordinates of the representative point of the currently touched square as the answer route information 532 (step S146), displays the route indicator 32 so that the route indicator 32 extends to the representative point of the currently touched square from the representative point of the preceding square along the answer route (step S148; see FIG. 3), and finishes the answer route display process.

[0212] When the processing section 200 has determined that the currently touched square is a square preceding the final square along the answer route based on the answer route information 532, the processing section 200 determines that the player has returned to the preceding square along the answer route (YES in step S144). The processing section 200 then deletes the registration information about the currently touched square from the answer route information 532 (step S150), displays the route indicator 32 so that the route indicator 32 is shortened to the representative point of the preceding square along the answer route (step S152), and finishes the answer route display process.

[0213] As shown in FIG. 22, when the processing section 200 has not detected that the player has touched the answer completion button 16 (i.e., has not detected the answer completion operation) (NO in step S170), the processing section 200 stops measuring the play time 534 and updating the time limit 8 (step S174), and generates the answer character string 546 based on the answer route information 532 (step S176). Specifically, the processing section 200 refers to the character link data 526b included in the question array data 526, and sequentially arranges the characters corresponding to the touched square representative point coordinates 532d in the touch order 532a included in the answer route information 532 to generate the answer character string 546 (see FIGS. 19 and 20).

[0214] When the processing section 200 has detected that the player has touched the answer completion button 16 (i.e., has detected the answer completion operation) (YES in step S170), the processing section 200 stops measuring the play time 534 and updating the time limit 8 (step S174), and generates the answer character string 546 based on the answer route information 532 (step S176). Specifically, the processing section 200 refers to the character link data 526b included in the answer route information 532 to generate the answer character string 546 (see FIGS. 19 and 20).

[0215] When the processing section 200 has determined that the answer character string 546 coincides with the right character string 524 (YES in step S180), the processing section 200 increments the successive right answer count 544 by one (step S182). When the processing section 200 has determined that the successive right answer count 544 has satisfied a given level increment condition (YES in step S184), the processing section 200 increments the difficulty level 520 by one (step S186). Note that the parameter (condition) for increasing the difficulty level 520 is not limited to the successive right answer count 544, but may be the number of right answers, the total acquired points, or the like.
[0216] The processing section 200 then calculates basic points using a given function based on the play time 534 that is not currently measured (step S194), and subtracts points from the basic points based on the hint disclosure count 530 to calculate the currently acquired points 550 (step S196). The processing section 200 then performs a clear effect display process (step S198).

[0217] FIG. 26 is a flowchart illustrative of the flow of the clear effect display process according to this embodiment. The processing section 200 displays the moving object 34 on the starting point end of the route indicator 32 (step S210; see FIG. 4), and changes the display state of the starting point square (step S212). In this embodiment, the processing section 200 changes the display state of the starting point square so that the square bursts into flame (moving object 34).

[0218] The processing section 200 moves the moving object 34 along the route indicator 32, and deletes the area of the route indicator 32 through which the moving object 34 has passed (step S214). In this embodiment, the processing section 200 controls the display state so that the flame (moving object 34) advances along the route indicator 32. When the processing section 200 has determined that the moving object 34 has moved to another square (YES in step S216), the processing section 200 changes the display state of the other square in the same manner as in the step S212 (step S218).

[0219] The above process is repeated until the moving object 34 reaches the end of the route indicator 32 (NO in step S220). Therefore, an effect image in which the flame advances along a fuse so that the squares are sequentially burned is displayed (see (1) and (2) in FIG. 4).

[0220] When the processing section 200 has determined that the moving object 34 has reached the end of the route indicator 32 (YES in step S220), the processing section 200 reads a prize image corresponding to the currently acquired points 550 from the prize image library 510 (step S222), causes the selected prize image to appear from behind the squares for which the display state has been changed (step S224; see (3) in FIG. 4), and finishes the clear effect display process.

[0221] As shown in FIG. 22, the processing section 200 then updates the total acquired points 552 (step S240), displays the advance operation button 38 and the game finish button 40 within the game screen instead of the hint button 14 and the answer completion button 16, and waits for the player to perform the next selection operation (step S242).

[0222] When the processing section 200 has detected that the player has touched the game finish button 40 (“FINISH” in step S244), the processing section 200 finishes the process. When the processing section 200 has not detected that the player has touched the game finish button 40 (“NEXT QUESTION” in step S244), the processing section 200 returns to the step S4.

[0223] When the processing section 200 has determined that the play time 534 has reached a given time limit (YES in step S172), and has determined that the answer character string 546 does not coincide with the right character string 524 in the step S180 (NO in step S180), the processing section 200 clears the successive right answer count 544 to “0” (step S250), and performs a failure effect display process (step S252).

[0224] FIG. 27 is a flowchart illustrative of the flow of the failure effect display process according to this embodiment. The processing section 200 refers to the answer route information 532, and extracts the square at which the answer route initially deviates from the right answer route as a failure square (step S270). Specifically, the processing section 200 refers to the character link data 526 included in the question array data 526 and the answer route information 532, sequentially compares the characters linked to the touched square representative point coordinates 532a along the answer route with the corresponding characters of the right character string 524 in the touch order 532a, and extracts the square at which the answer route initially deviates from the right answer route as a failure square.

[0225] The processing section 200 then displays the moving object 34 on the starting point end of the route indicator 32 (step S272), and changes the display state of the starting point square (step S274). The processing section 200 moves the moving object 34 along the route indicator 32, and deletes the area of the route indicator 32 through which the moving object 34 has passed (step S276).

[0226] When the processing section 200 has determined that the moving object 34 has moved to another square (YES in step S278), the processing section 200 changes the display state of the other square (step S280). The above process is repeated until the moving object 34 reaches the failure square (NO in step S282). Therefore, an effect image in which the flame advances along a fuse so that the squares are sequentially burned is initially displayed (see (1) in FIG. 5).

[0227] When the processing section 200 has determined that the moving object 34 has reached the failure square (YES in step S282), the processing section 200 causes the moving object 34 to disappear (step S284). In this embodiment, an effect image in which the flame goes out is displayed (see (2) in FIG. 5).

[0228] The processing section 200 then displays a given failure notification text in the message display area 10 (step S286; see (3) in FIG. 5), and finishes the failure effect display process.

[0229] As shown in FIG. 22, the processing section 200 then displays the game finish button 40 and the retry button 42 within the game screen instead of the hint button 14 and the answer completion button 16, and waits for the player to perform the next selection operation (step S302).

[0230] When the processing section 200 has detected that the player has touched the retry button 42 (“RETRY” in step S302), the processing section 200 returns to the step S100. When the processing section 200 has detected that the player has touched the game finish button 40 (“FINISH” in step S302), the processing section 200 finishes the process.

[0231] According to this embodiment, it is possible to implement a puzzle game that disposes characters of a character string (e.g., character, numeral, or sign) having a meaning respectively in a plurality of adjacent disposed squares, and allows the player to input an answer route by connecting all or some of the squares along a one-way route to find the character string.

[0232] The answer route can be displayed so that the player can easily determine the answer route visually, and whether or not the answer route is right can be displayed in an elaborate way. When the player has desired a hint for finding the answer route, the hint can be displayed. According to this
embodiment, a question can be set while setting the difficulty level from various viewpoints.

Second Embodiment

[0233] A second embodiment to which the invention is applied is described below. This embodiment is basically implemented by the same hardware configuration as that of the first embodiment, but differs from the first embodiment in that the right character string can be acquired from data other than the right character string set 514. The following description focuses on the difference from the first embodiment. The same elements as those of the first embodiment are indicated by identical reference symbols. Description of these elements is omitted.

[0234] Functional Configuration

[0235] FIG. 28 is a functional block diagram showing an example of the functional configuration according to this embodiment. This embodiment is basically implemented by the same functional configuration as that of the first embodiment. In this embodiment, however, the processing section 200 includes an e-mail control section 230 that sends and receives an e-mail (electronic mail) by reading and executing an e-mail program 570 from the storage section 500. Data sent or received via an e-mail is stored in the storage section 500 as mail data 572 (i.e., private database). The mail data 572 may in a Multipurpose Internet Mail Extensions (MIME) format, a dbx format, an HTM L format, or the like.

[0237] A game program 502B accordingly to this embodiment includes a mail data decoder program 574 that decodes the mail data 572 into text data. The processing section 200 includes a decoder section 232 that appropriately reads and executes the mail data decoder program 574, and converts the mail data 572 into text data during the game.

[0238] Process Flow

[0239] The flow of the process according to this embodiment is basically the same as that of the first embodiment. In this embodiment, however, a right character string selection process is performed instead of the process in the steps S4 to S8.

[0240] FIG. 29 is a flowchart illustrative of the flow of the right character string selection process. The processing section 200 performs a lottery process that determines whether or not to extract the right character string from the mail data 572 (external data) (step S320). For example, an appropriate win probability is assigned using a known lottery process that generates random numbers.

[0241] When it has been determined to extract the right character string from the external data (YES in step S322), the processing section 200 acquires the number-of-character condition 514d corresponding to the current difficulty level 520 from the right character string set 514 (step S324; see FIG. 17).

[0242] The processing section 200 then reads and executes the mail data decoder program 574, and decodes the mail data 572 to obtain a text file (step S326). The processing section 200 refers to the word dictionary data 518, and extracts a single phrase or a range of phrases that satisfies the acquired number-of-character condition from the text file (step S328). The processing section 200 then selects one of the extracted phrase and range of phrases, and uses the selected phrase as the right character string 524 (step S330). The processing section 200 sets the right character string ID 522 to a given value "NULL" or the like that indicates that the right character string set 514 does not include the corresponding ID (step S332), and finishes the right character string selection process.

[0243] When it has been determined not to extract the right character string from the external data (NO in step S322), the processing section 200 selects the right character string set 514 corresponding to the current difficulty level 520 in the same manner as in the steps S4 to S8 in the first embodiment (step S340), and selects the right character string 514e from the selected right character string set 514 (step S342). The processing section 200 then stores the corresponding right character string ID 514d as the right character string ID 522 (step S344), determines the selected right character string 514e to be the right character string 524 (step S346), and finishes the right character string selection process.

[0244] According to this embodiment, a word, a phrase, a secret language, or the like used by the player, his friend, or the like can be used as the question.

[0245] Note that the right character string may not be extracted from the mail data 572. The right character string may be extracted from a private database (e.g., directory, telephone book, diary, or schedule book) that can be edited by the player by performing an operation input (e.g., HTML file including text data or document data). In this case, a decoder program for such a private database is provided as the game program.

[0246] When the right character string ID 522 is "NULL", a hint is not disclosed even if the player has performed the hint disclosure operation, or a message "refer to mail data" is displayed when the mail data 572 is referred to, or the subject of the mail is displayed as a hint in the steps S110 to S114 (see FIG. 21). When using a private database other than mail data, the name (telephone book or directory) of the database is displayed as a hint, or the file name of the database is displayed as a hint, for example.

MODIFICATIONS

[0247] The embodiments to which the invention is applied have been described above. Note that the invention is not limited thereto. Various modifications may be appropriately made, such as adding other elements, omitting some of the elements, or changing some of the elements.

First Modification

[0248] For example, the game device may be implemented by a stationary consumer game device or an arcade game device instead of the portable game device 1400.

[0249] An electronic instrument used as the game device is not limited to an electronic instrument sold as a game device. An electronic instrument (e.g., mobile phone, compact digital camera, music player, personal computer, or car navigation system) that can execute application software is considered to include a computer that can execute a program, or considered to be a computer. Therefore, such an electronic instrument may be used as the game device that implements the invention.

[0250] As shown in FIG. 30, a portable terminal 1500 that includes a touch panel 1509 that covers the display area of a liquid crystal display 1508 may be used as the game device, for example. The portable terminal 1500 includes a speaker 1510, a microphone 1511, and a radio communication device used to communicate with a base station, and functions as a mobile phone or a PHS. The portable terminal 1500 reads an
application program from a memory card 1540 via a memory card reader 1518, or downloads an application program from an external device by connecting to a communication line 1, and executes the application program using a processor such as CPU mounted on a control board 1550. The portable terminal 1500 includes an arrow key 1502, and may be used to input an answer route from the starting point square. The portable terminal 1500 reads the game program 502 and various types of data from the memory card 1540 or the like, and thus functions as the game device that implements the invention.

Second Modification

[0251] The above embodiments have been described taking an example in which the prize image is an effect image that merely indicates the currently acquired points. Note that the prize image is not limited thereto. For example, the prize image may be an image that relates to the next question, or may be an image including a hint for finding the right answer route.

[0252] In this case, a hint image is included in the right character string set 514 in addition to the right character string ID514d and the right character string 514e. In the example shown in FIG. 17, when the right character string 514e is “Takamakii (tuna roll)”, the hint image may be an image of pickled ginger, nori, or the like.

[0253] The process flow is changed as follows. Specifically, a step that selects one right character string set 514 based on the current difficulty level 520, a step that selects one right character string 514e from the selected right character string set 514, and a step that stores the right character string ID 514d of the selected right character string 514e as the right character string set 522 are added between the step S186 and the step S194 (see FIG. 22).

[0254] In the clear display effect display process (see FIG. 26), the processing section 200 performs a step that reads the hint image corresponding to the next right character string ID522 from the right character string set 514, and a step that displays the hint image so that the hint image appears from behind the squares that have changed in display state instead of the steps S222 and S224.

[0255] When the processing section 200 has detected that the player has touched the advance operation button in the step S244 (see FIG. 22) (“NEXT QUESTION” in step S244), the processing section 200 returns to the step S10 instead of the step S4.

Third Modification

[0256] In the question array generation process according to the above embodiments, a plurality of pieces of array model data 516 are provided in advance, and the array model 516c that satisfies the question condition and the array model data 516 that satisfies the arrangement order 516e are selected. Note that the question array generation process is not limited thereto.

[0257] When the array model data 516 is not used, for example, the processing section 200 sequentially and adjacently disposes squares in the same number as the number of characters of the right character string 524 and the value set as the question condition “unnecessary character” in a random direction to generate the array model (step S33) instead of performing the step S34 according to the first embodiment (see FIGS. 31 and 32). When the question condition “unnecessary character” is set a value other than “0”, the processing section 200 sequentially and adjacently disposes squares in the same number as the total of the number of characters of the right character string 524 and the value set as the question condition “unnecessary character” in a random direction to generate the array model (step S37) instead of performing the step S36 according to the first embodiment. Specifically, information that corresponds to the array model 516c and the arrangement order 516e according to the first embodiment are automatically generated without using the array model 516c and the arrangement order 516e defined by the array model data 516.

[0258] The processing section 200 then determines whether or not the array model generated in the step S33 or S37 satisfies each of the remaining question conditions instead of performing the step S40 according to the first embodiment (step S41). When the processing section 200 has determined that the array model does not satisfy each of the remaining question conditions (NO in step S41), the processing section 200 discards the array model instead of performing the step S42 according to the first embodiment (step S43), returns to the step S32, and generates the array model again.

[0259] The step S88 is replaced by a step S89 so that the question array model is generated based on the generated array model 516c and the arrangement order 516e.

Fourth Modification

[0260] The clear effect image and the failure effect image may be appropriately changed.

[0261] FIG. 33 is a view showing a modification of the clear effect image. As shown in FIG. 33 (see (1)), a bait 50 may be displayed in each square of the route indicator 32 so that the moving object 34 sequentially eats the bait 50 from the starting point. In this case, the square through which the moving object 34 has passed may be changed into a transparent body as if the moving object 34 had eaten the square. When the moving object 34 has reached the end of the route indicator 32 (see (2) in FIG. 33), a favorite food 52 of the moving object 34 is displayed within the screen, and a state in which the moving object 34 runs after the favorite food 52 with pleasure may be displayed comically. This improves the effect.

[0262] As a modification of the failure effect image, a special bait (hard and large bait) differing from the bait 50 may be displayed in the failure square, and a state in which the moving object 34 eats the special bait with considerable effort may be displayed. A predator that eats the moving object 34 may be displayed within the game screen, and a state in which the predator eats runs after the moving object 34 may be displayed.

Other Modifications

[0263] The above embodiments have been described taking an example in which the right character string set 514 and the array model data 516 are stored in the storage section 500 in advance. Note that the right character string set 514 and the array model data 516 may be downloaded from an external device (e.g., a server device of the game manufacturer) via the communication section 370.

[0264] The shape of the square is not limited to a quadrilateral. As shown in FIGS. 34 and 35, the square may have another polygonal shape. The squares included in one question array model may differ in shape and/or size. The square may be a figure enclosed by a closed curve (e.g., piece of a jigsaw puzzle). The above embodiments have been described
taking an example in which one character is assigned to each square (i.e., the number of assignment characters is one). Note that the question array model may be generated to include a square to which two or more characters are assigned (see (2) in FIG. 35). Two or more characters may be assigned to each square.

[0265] The above embodiments have been described taking an example in which, when a natural number is set as the question condition “unnecessary character”; a square to which a character irrelevant to the right character string is assigned is included in the array model in which the number of squares is obtained by adding the natural number set as the question condition “unnecessary character” to the number of characters of the right character string without changing the character string of the right character string. Note that another configuration may also be employed. For example, the array model may be updated by adding (inserting) a given sign (unnecessary sign) that is considered to be absent even if it is inserted into the right character string.

[0266] Although only some embodiments of the invention have been described in detail above, those skilled in the art would readily appreciate that many modifications are possible in the embodiments without materially departing from the novel teachings and advantages of the invention. Accordingly, such modifications are intended to be included within the scope of the invention.

What is claimed is:

1. A game execution method comprising:
   displaying a question array generated by adjacent disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
   inputting an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
   comparing an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

2. The method according to claim 1, further comprising: discriminately displaying squares among the plurality of squares along the right route.

3. The method according to claim 1, further comprising: displaying a moving object so that the moving object traces squares among the plurality of squares along the answer route.

4. The method according to claim 3, further comprising: changing a display state of each square among the plurality of squares through which the moving object moves.

5. The method according to claim 3, further comprising: displaying a new image hidden by the question array when it has been determined that the answer character string is correct, and the moving object has reached a final square along the answer route.

6. The method according to claim 1, further comprising: displaying a hint based on the right character string.

7. The method according to claim 1, further comprising: displaying a hint by displaying the meaning of the right character string.

8. The method according to claim 1, further comprising: displaying a hint by displaying a first assignment character and/or a final assignment character of the right character string.

9. The method according to claim 1, further comprising: displaying a hint by displaying a word included in the right character string.

10. The method according to claim 1, further comprising: performing a given effect process based on an elapsed time from a given start timing.

11. The method according to claim 1, further comprising: performing a filter effect display process that gradually decreases the visibility of each of the plurality of squares based on an elapsed time from a given start timing.

12. The method according to claim 1, further comprising: calculating points based on at least an elapsed time from a given start timing.

13. The method according to claim 6, further comprising: calculating points based on whether or not the hint has been displayed.

14. The method according to claim 1, further comprising: determining the right character string by reading characters, numerals, signs, or a combination thereof having a meaning from a private database that is selected from a directory, a telephone book, a diary, a schedule book, and e-mail data and can be edited by a player by performing an operation input.

15. A non-transitory storage medium storing a program that causes a computer to execute the method according to claim 1.

16. An electronic instrument comprising:
   a question array display control section that displays a question array generated by adjacent disposing a plurality of squares, an assignment character that is at least one character, numeral, sign, or a combination thereof being assigned to each of the plurality of squares, and a right character string having a meaning being obtained by tracing squares among the plurality of squares along a right route so that each of the plurality of squares is passed only once;
   an answer route input section that inputs an answer route that traces squares among the plurality of squares included in the question array based on an operation input performed by a player; and
   a right/wrong determination section that compares an answer character string obtained by arranging the assignment characters along the answer route with the right character string to determine whether or not the answer character string is correct.

* * * * *