US 20100105456A1

(19) **United States**

(12) **Patent Application Publication**

Marufuji

(10) **Pub. No.: US 2010/0105456 A1**

(43) **Pub. Date: Apr. 29, 2010**

(54) **GAME TERMINAL, GAME SYSTEM THEREFOR, COMPUTER PROGRAM THEREFOR, AND COMPUTER READABLE RECORDING MEDIUM THEREWITH**

(76) Inventor: **Yohei Marufuji**, Tokyo (JP)

Correspondence Address:
WINSTON & STRAWN LLP
PATENT DEPARTMENT
1700 K STREET, N.W.
WASHINGTON, DC 20006 (US)

(21) Appl. No.: **12/523,374**

(22) PCT Filed: **Jan. 17, 2008**

(86) PCT No.: **PCT/JP2008/050497**

§ 371 (c)(1),
(2), (4) Date: **Sep. 9, 2009**

(30) **Foreign Application Priority Data**

Jan. 19, 2007 (JP) 2007-010246
Jan. 23, 2007 (JP) 2007-012242

Publication Classification

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

(52) **U.S. Cl.** **463/9**; 463/43; 463/42

(57) **ABSTRACT**

The present invention provides a game terminal (**10**). In a quiz, processor (**150**) of game terminal (**10**) performs a process of asking a quiz question, receiving an answer to the quiz question, and notifying a result of the quiz after the quiz is completed (SD**6**, SD**10**, SD**12**, SD**14**, SF**3**, SF**9**, SF**10**, SF**11**), a process of converting answer data representing a received answer in such a manner that, when the answer is correct, the encoded bit string is the shortest (SD**7**, Sf**4**), a process of transmitting the converted answer data to another game terminal (**10**) (SD**9**, SF**5**), and a process of reverse-converting the answer data after receiving the answer data from another game terminal (**10**) SD**11**, SD**13**, SF**8**, SF**14**).

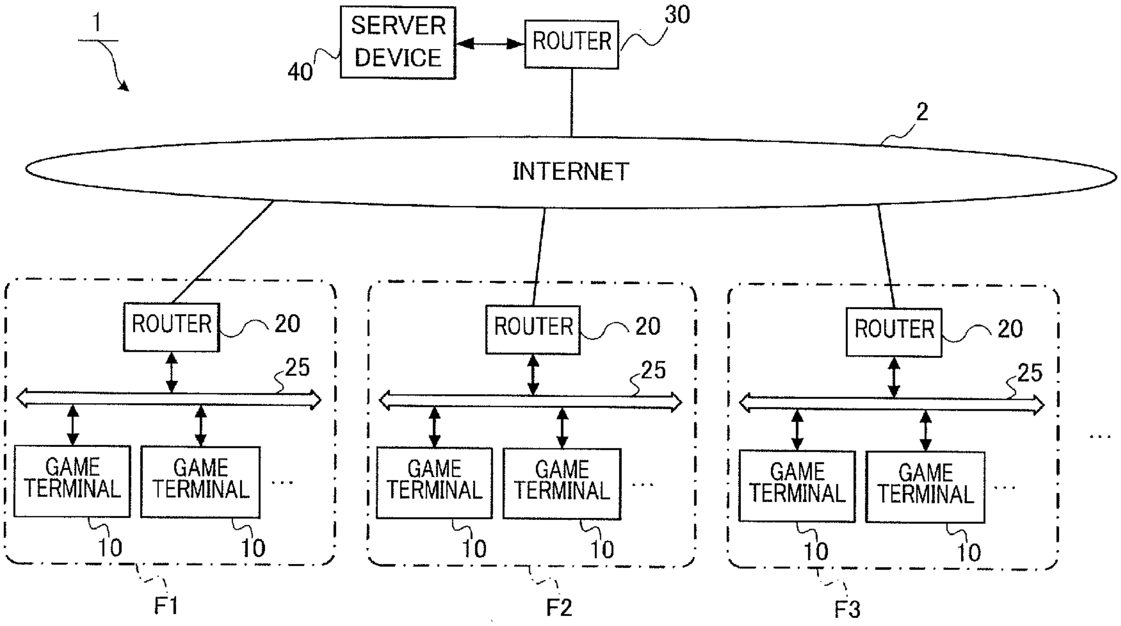


FIG. 1

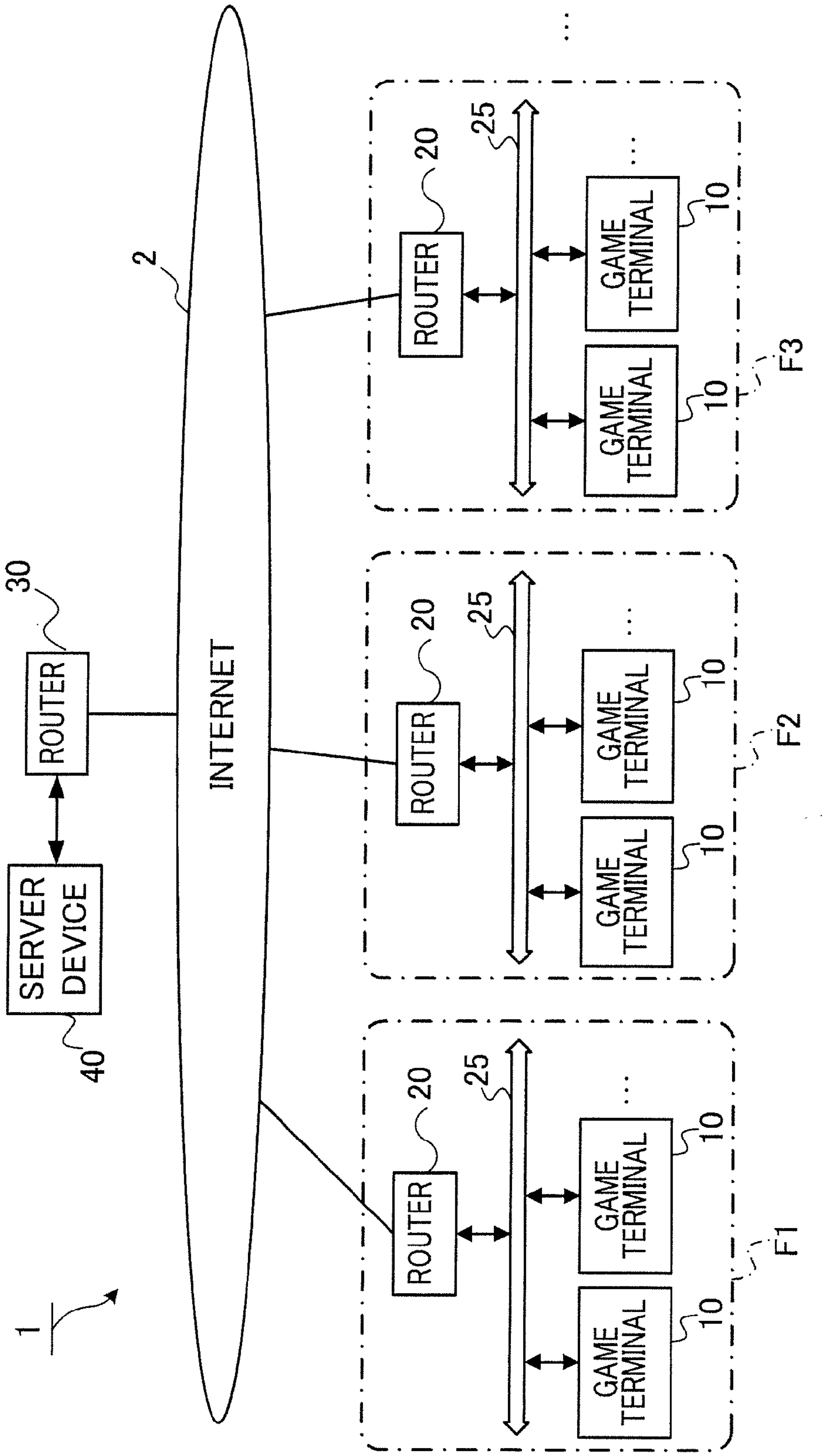

1
2
INTERNET
30
ROUTER
SERVER DEVICE
40
F1
F2
F3
ROUTER
20
25
GAME TERMINAL
10
GAME TERMINAL
10
ROUTER
20
25
GAME TERMINAL
10
GAME TERMINAL
10
ROUTER
20
25
GAME TERMINAL
10
GAME TERMINAL
10

FIG. 2

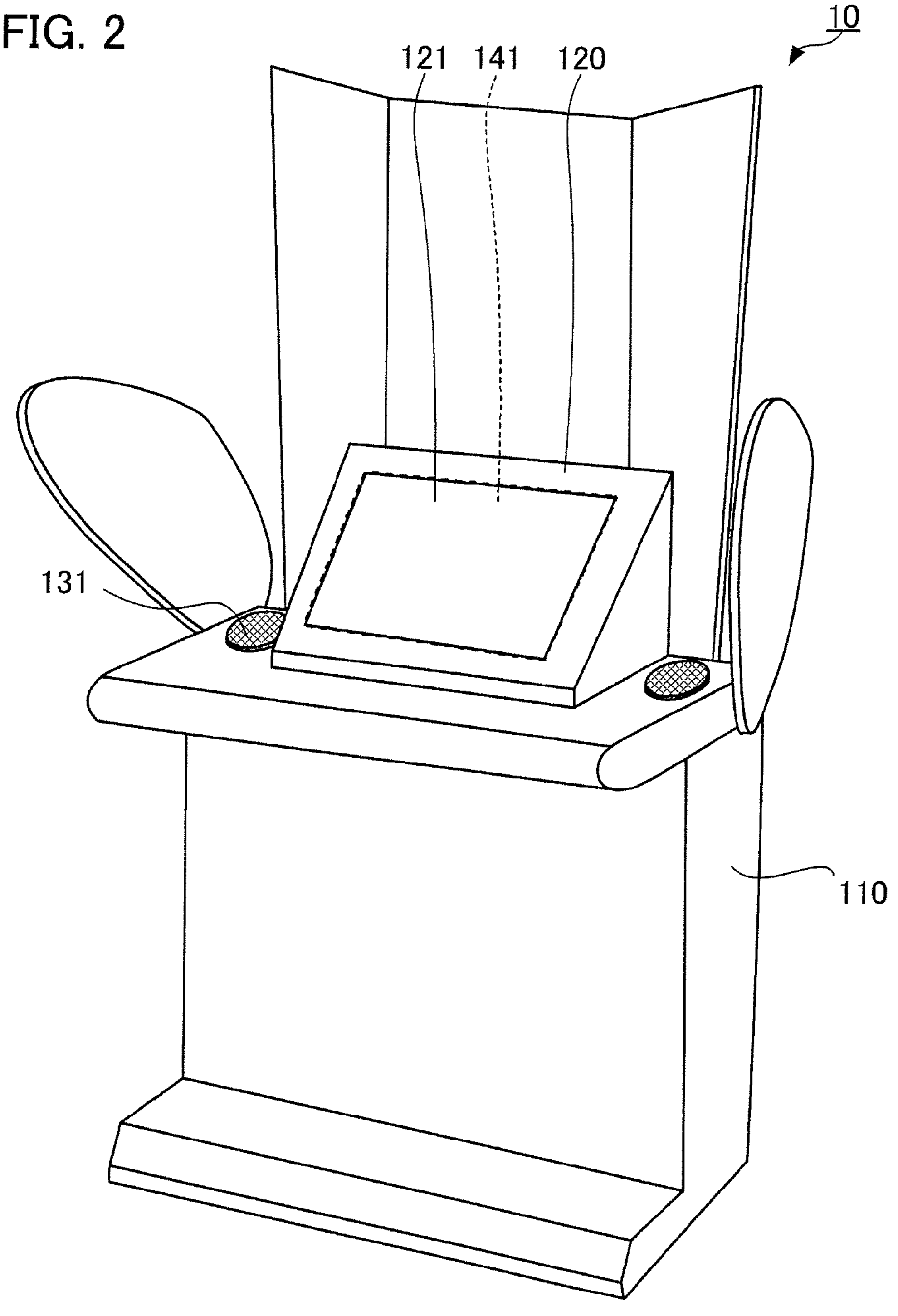


FIG. 3

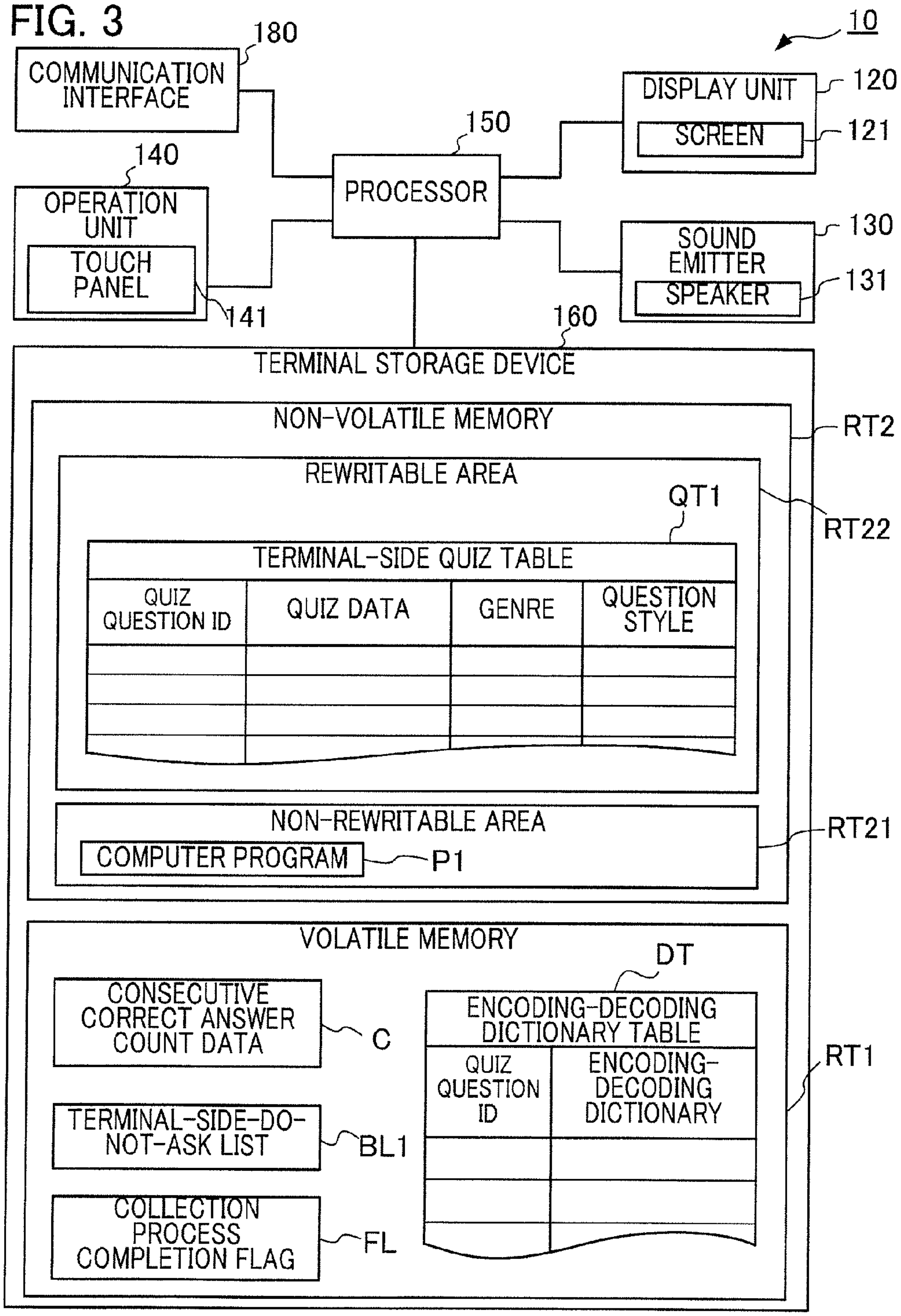

10
180
COMMUNICATION INTERFACE
150
PROCESSOR
DISPLAY UNIT
120
SCREEN
121
140
OPERATION UNIT
TOUCH PANEL
141
SOUND EMITTER
130
SPEAKER
131
160
TERMINAL STORAGE DEVICE
NON-VOLATILE MEMORY
RT2
REWRITABLE AREA
RT22
QT1
TERMINAL-SIDE QUIZ TABLE
QUIZ QUESTION ID
QUIZ DATA
GENRE
QUESTION STYLE
NON-REWRITABLE AREA
RT21
COMPUTER PROGRAM
P1
VOLATILE MEMORY
DT
CONSECUTIVE CORRECT ANSWER COUNT DATA
C
ENCODING-DECODING DICTIONARY TABLE
QUIZ QUESTION ID
ENCODING-DECODING DICTIONARY
RT1
TERMINAL-SIDE-DO-NOT-ASK LIST
BL1
COLLECTION PROCESS COMPLETION FLAG
FL

FIG. 4

QUIZ DATA	CORRECT ANSWER DATA	QUIZ-QUESTION RELATED KEYWORD GROUP DATA	CORRECT-ANSWER RELATED KEYWORD GROUP DATA	INCORRECT-ANSWER RELATED KEYWORD GROUP DATA

FIG. 5

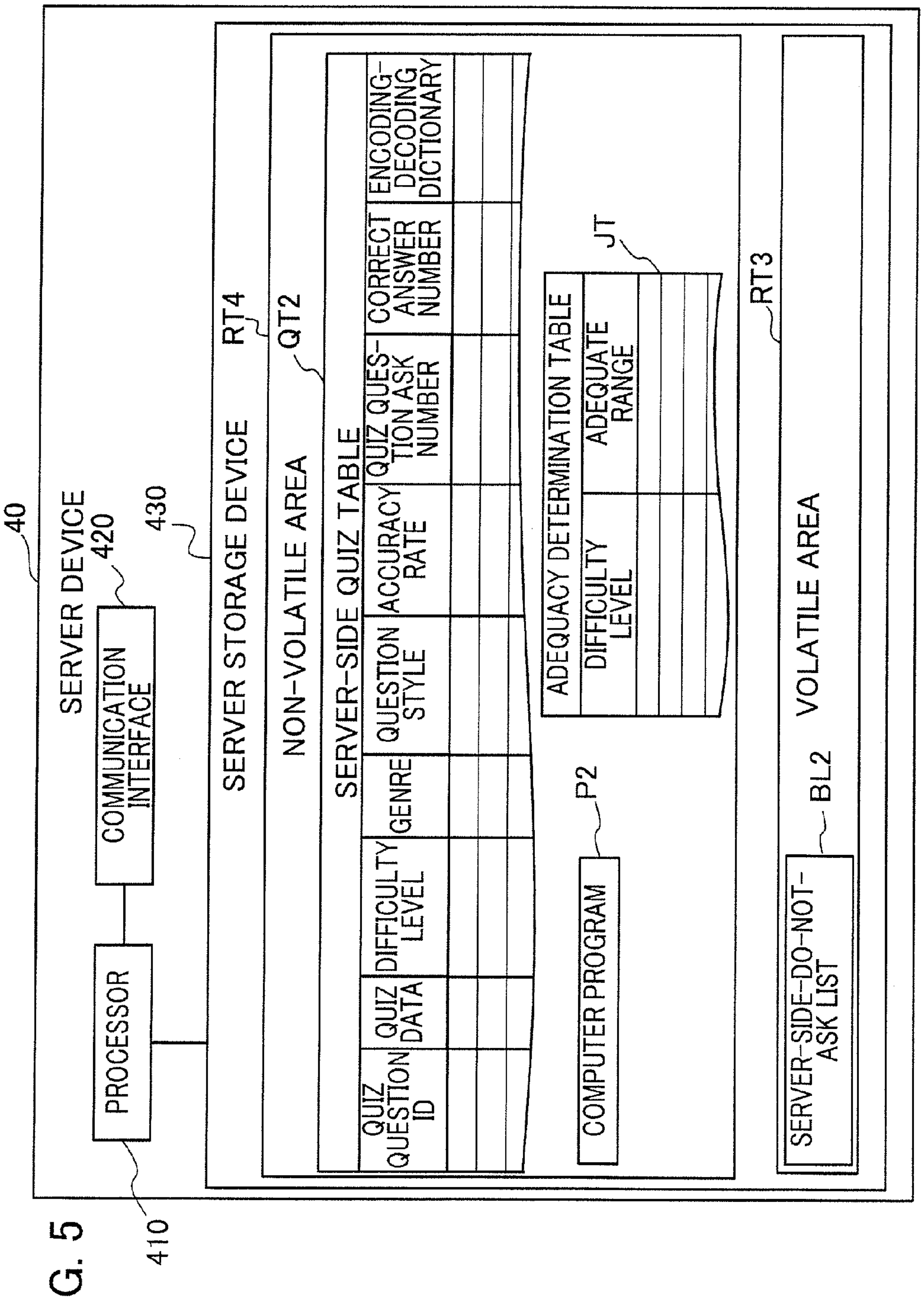

40
SERVER DEVICE
410
PROCESSOR
420
COMMUNICATION INTERFACE
430
SERVER STORAGE DEVICE
RT4
NON-VOLATILE AREA
QT2
SERVER-SIDE QUIZ TABLE
QUIZ QUESTION ID
QUIZ DATA
DIFFICULTY LEVEL
GENRE
QUESTION STYLE
ACCURACY RATE
QUIZ QUESTION ASK NUMBER
CORRECT ANSWER NUMBER
ENCODING-DECODING DICTIONARY
P2
COMPUTER PROGRAM
JT
ADEQUACY DETERMINATION TABLE
DIFFICULTY LEVEL
ADEQUATE RANGE
RT3
VOLATILE AREA
BL2
SERVER-SIDE-DO-NOT-ASK LIST

FIG. 6A

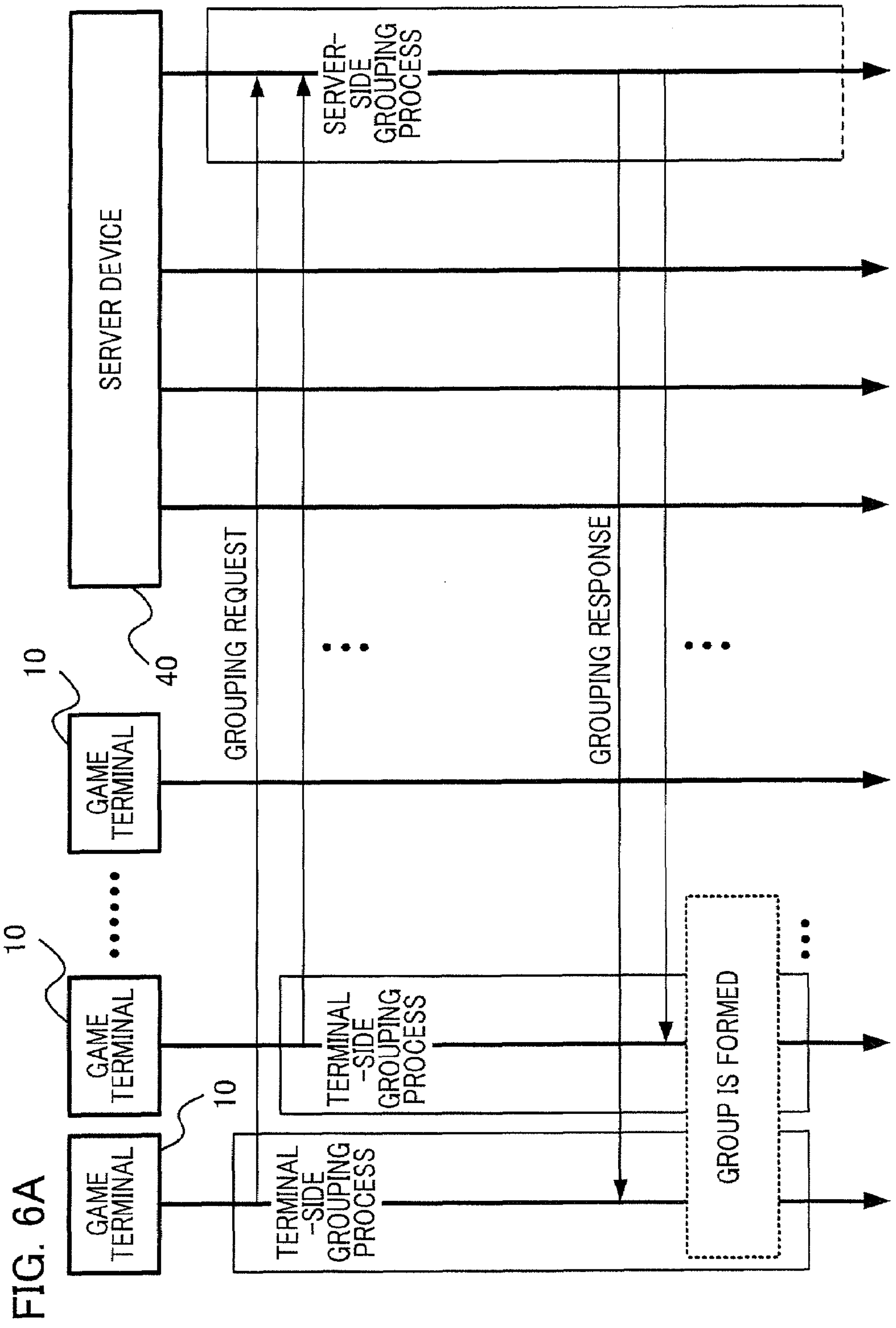

GAME TERMINAL
10
GAME TERMINAL
10
GAME TERMINAL
10
SERVER DEVICE
40
GROUPING REQUEST
GROUPING RESPONSE
SERVER-SIDE GROUPING PROCESS
TERMINAL-SIDE GROUPING PROCESS
TERMINAL-SIDE GROUPING PROCESS
GROUP IS FORMED

FIG. 6B

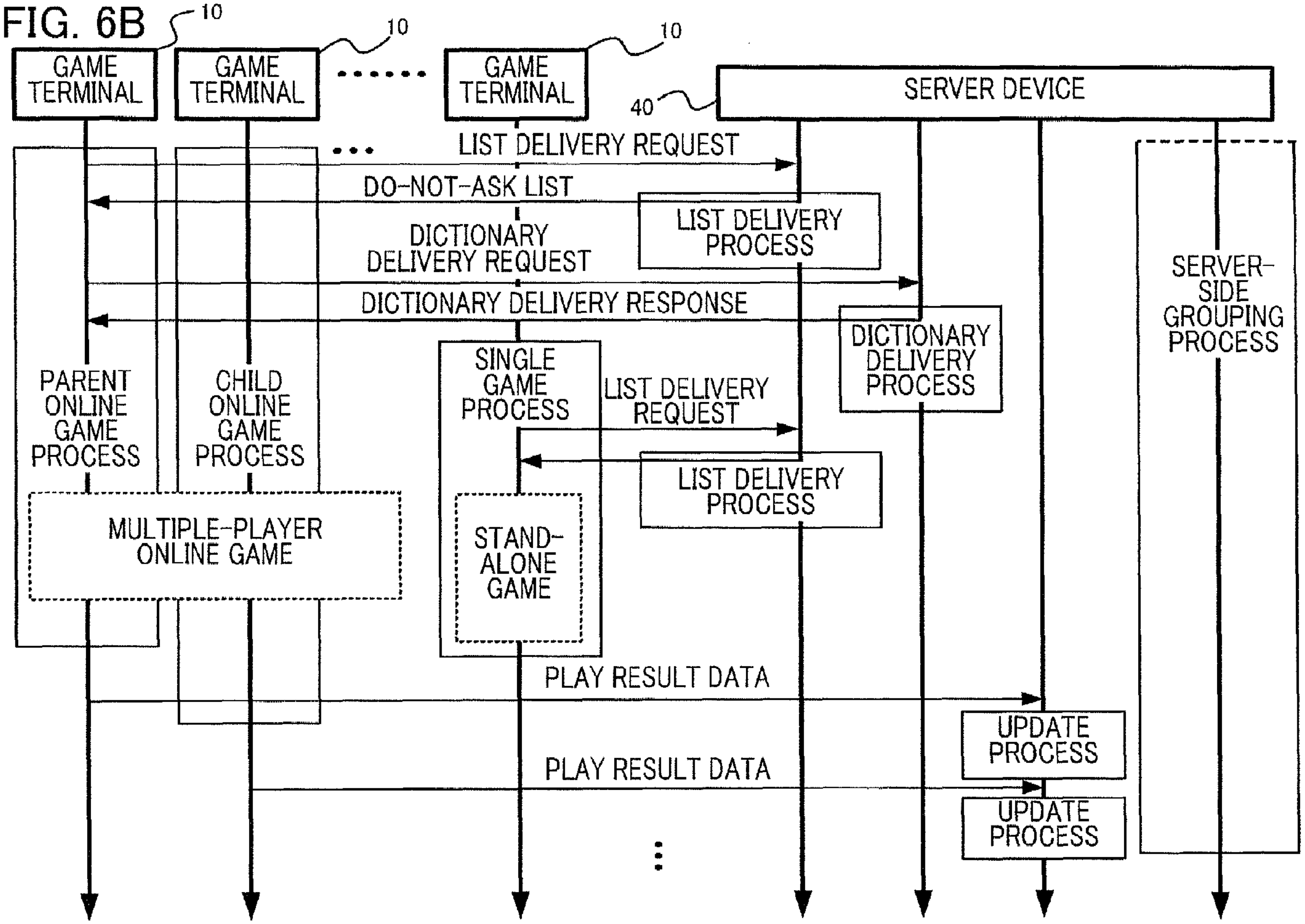

10
10
10
GAME TERMINAL
GAME TERMINAL
GAME TERMINAL
40
SERVER DEVICE
LIST DELIVERY REQUEST
DO-NOT-ASK LIST
LIST DELIVERY PROCESS
DICTIONARY DELIVERY REQUEST
DICTIONARY DELIVERY RESPONSE
DICTIONARY DELIVERY PROCESS
SERVER-SIDE GROUPING PROCESS
PARENT ONLINE GAME PROCESS
CHILD ONLINE GAME PROCESS
SINGLE GAME PROCESS
LIST DELIVERY REQUEST
LIST DELIVERY PROCESS
MULTIPLE-PLAYER ONLINE GAME
STAND-ALONE GAME
PLAY RESULT DATA
UPDATE PROCESS
PLAY RESULT DATA
UPDATE PROCESS

FIG. 7

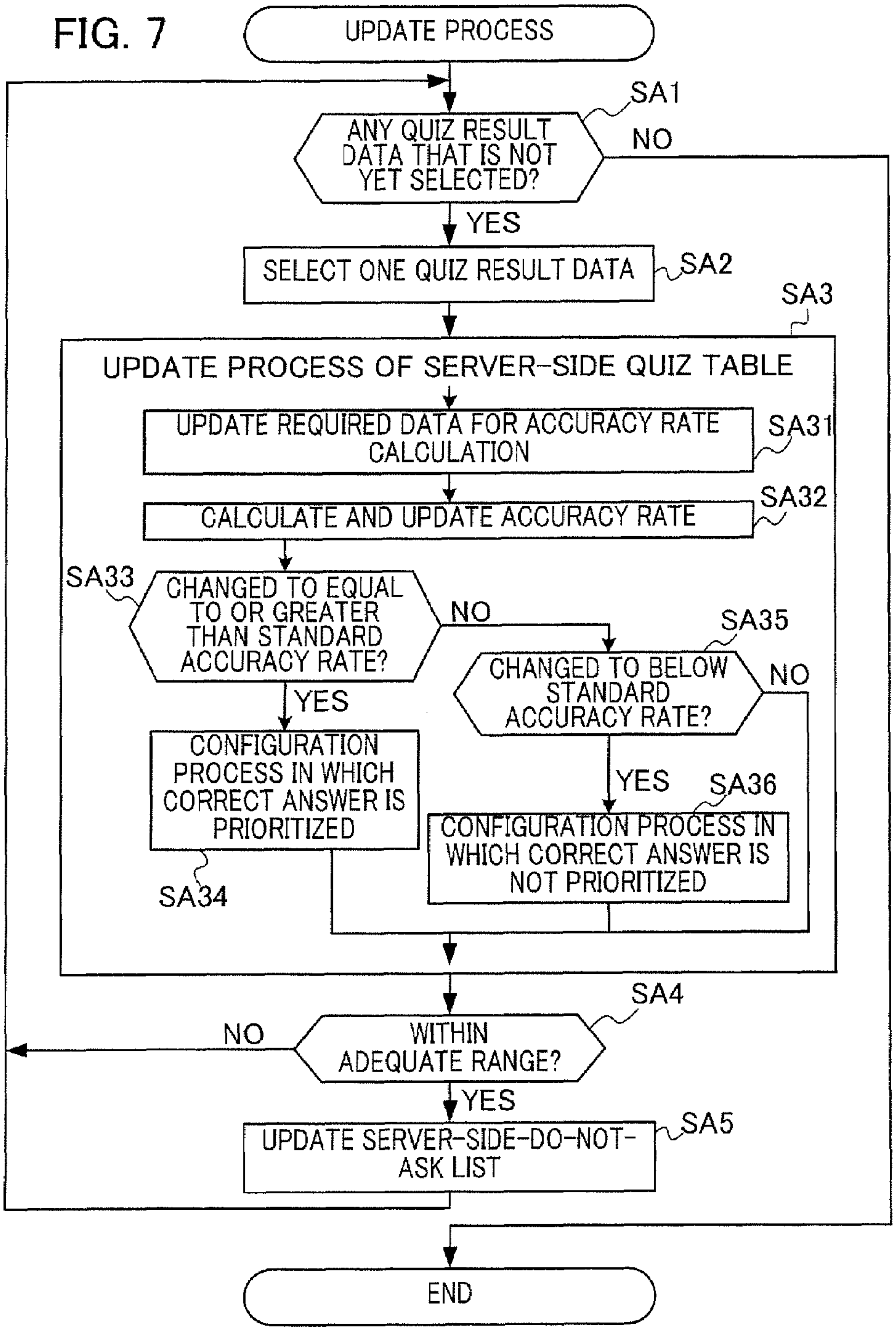

UPDATE PROCESS
SA1
ANY QUIZ RESULT DATA THAT IS NOT YET SELECTED?
NO
YES
SELECT ONE QUIZ RESULT DATA
SA2
SA3
UPDATE PROCESS OF SERVER-SIDE QUIZ TABLE
UPDATE REQUIRED DATA FOR ACCURACY RATE CALCULATION
SA31
CALCULATE AND UPDATE ACCURACY RATE
SA32
SA33
CHANGED TO EQUAL TO OR GREATER THAN STANDARD ACCURACY RATE?
NO
SA35
CHANGED TO BELOW STANDARD ACCURACY RATE?
NO
YES
CONFIGURATION PROCESS IN WHICH CORRECT ANSWER IS PRIORITIZED
YES
SA36
CONFIGURATION PROCESS IN WHICH CORRECT ANSWER IS NOT PRIORITIZED
SA34
SA4
NO
WITHIN ADEQUATE RANGE?
YES
UPDATE SERVER-SIDE-DO-NOT-ASK LIST
SA5
END

FIG. 8

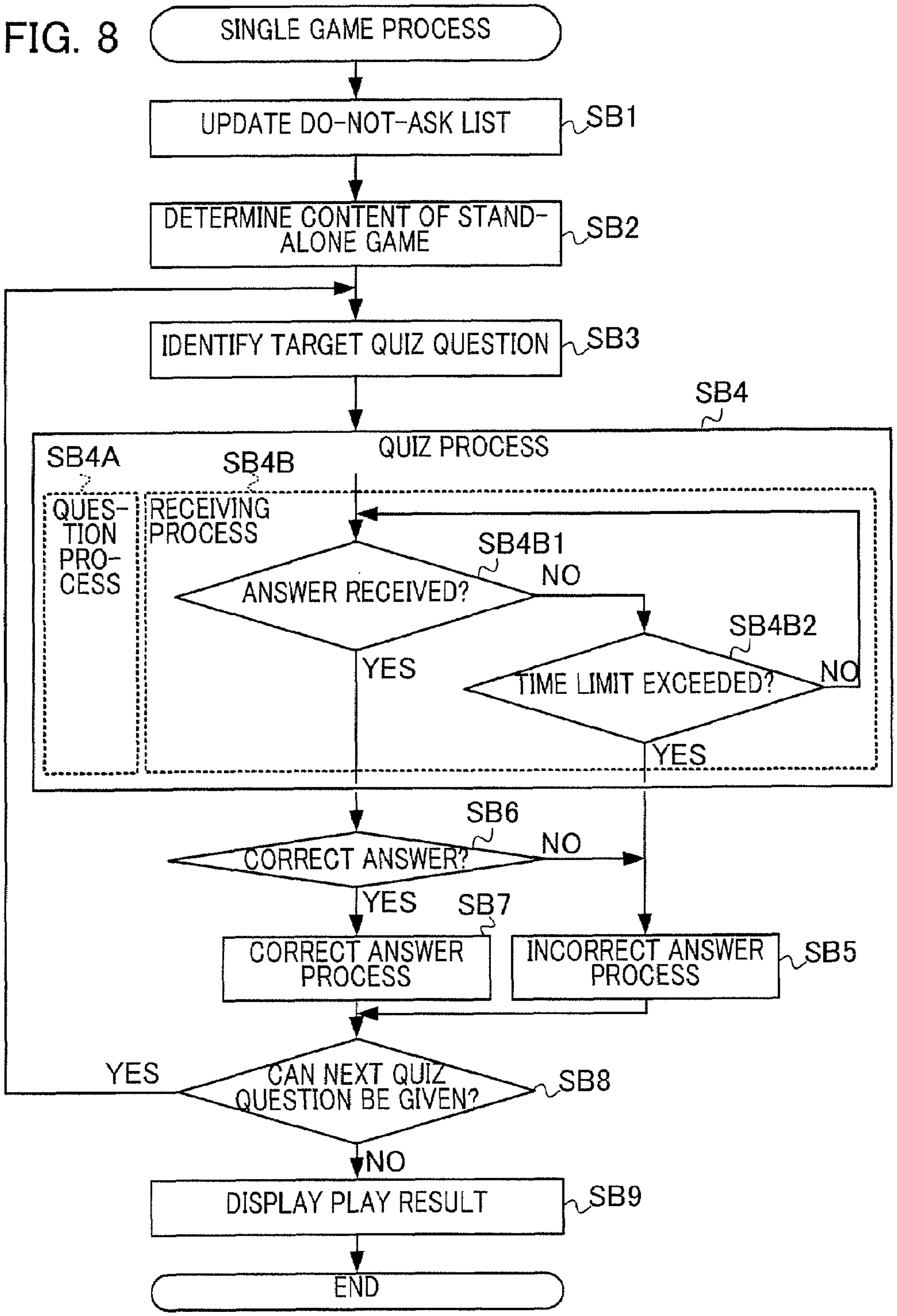

SINGLE GAME PROCESS
UPDATE DO-NOT-ASK LIST
SB1
DETERMINE CONTENT OF STAND-ALONE GAME
SB2
IDENTIFY TARGET QUIZ QUESTION
SB3
SB4
QUIZ PROCESS
SB4A
SB4B
QUESTION PROCESS
RECEIVING PROCESS
SB4B1
ANSWER RECEIVED?
NO
YES
SB4B2
TIME LIMIT EXCEEDED?
NO
YES
SB6
CORRECT ANSWER?
NO
YES
SB7
CORRECT ANSWER PROCESS
INCORRECT ANSWER PROCESS
SB5
YES
CAN NEXT QUIZ QUESTION BE GIVEN?
SB8
NO
DISPLAY PLAY RESULT
SB9
END

FIG. 9

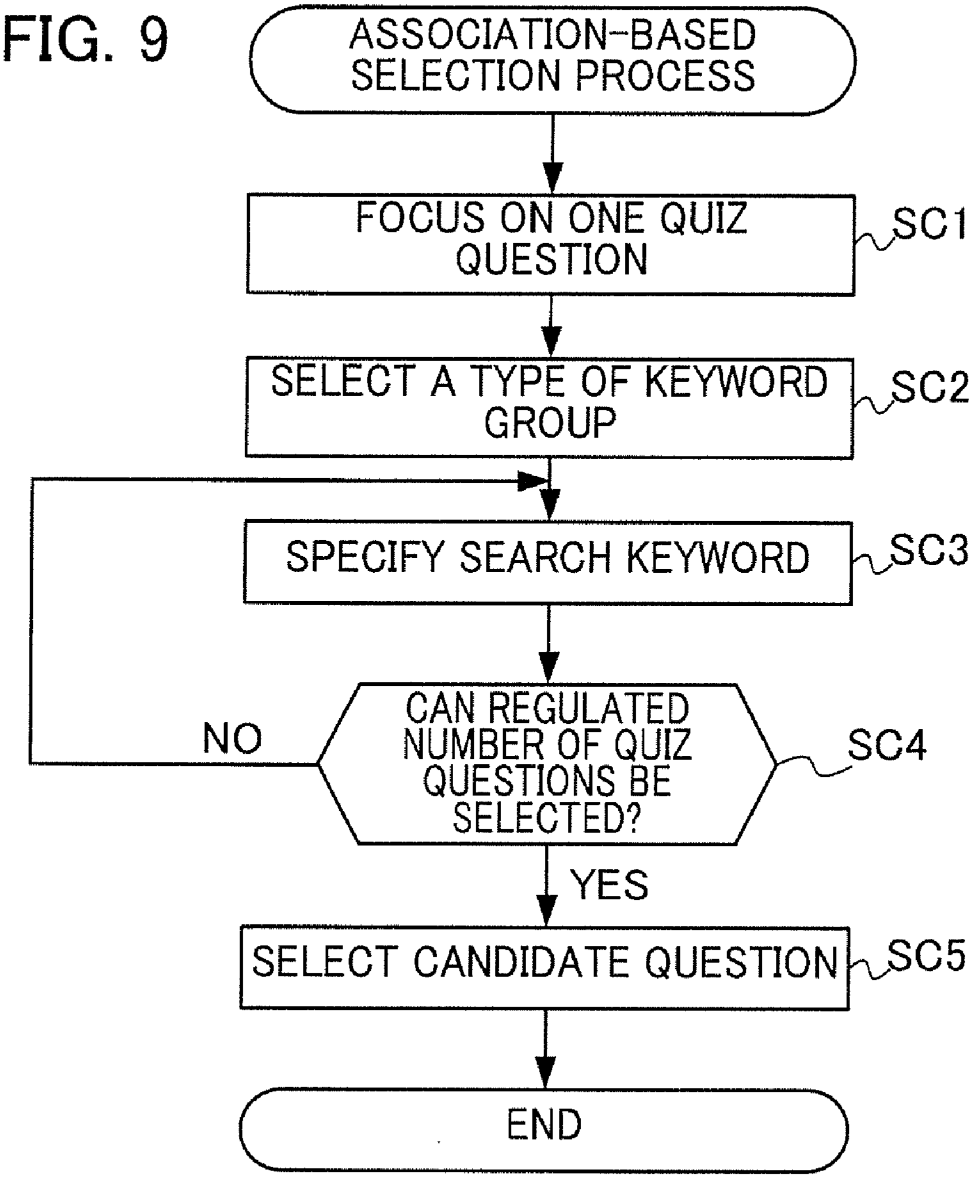

ASSOCIATION-BASED SELECTION PROCESS
FOCUS ON ONE QUIZ QUESTION
SC1
SELECT A TYPE OF KEYWORD GROUP
SC2
SPECIFY SEARCH KEYWORD
SC3
CAN REGULATED NUMBER OF QUIZ QUESTIONS BE SELECTED?
SC4
NO
YES
SELECT CANDIDATE QUESTION
SC5
END

FIG. 10

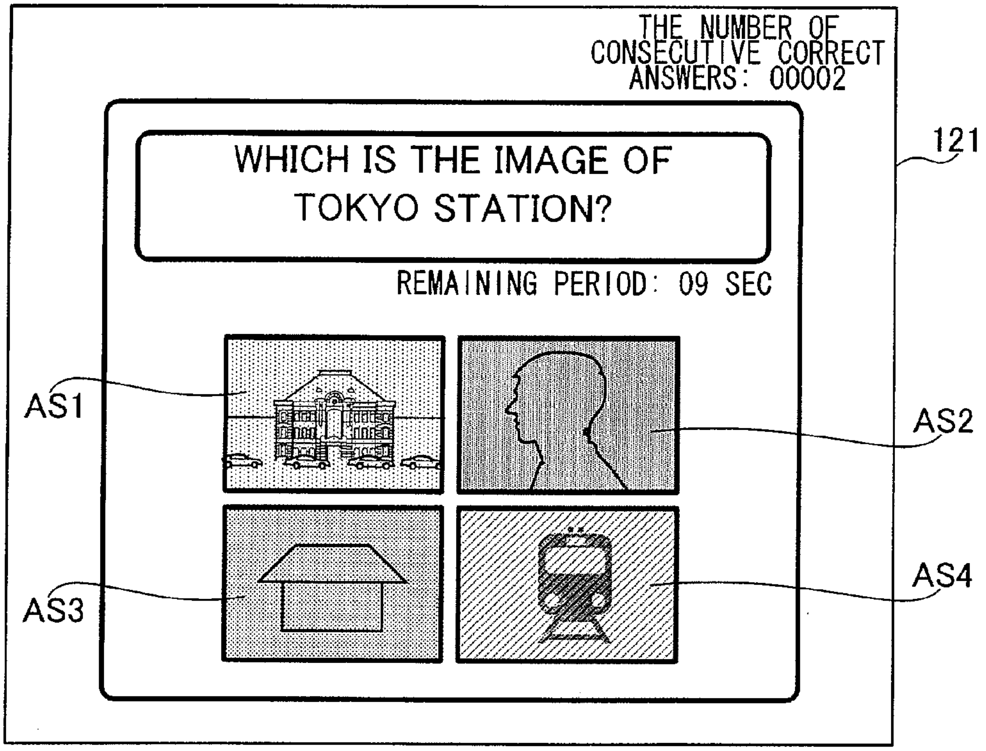

THE NUMBER OF
CONSECUTIVE CORRECT
ANSWERS: 00002
121
WHICH IS THE IMAGE OF
TOKYO STATION?
REMAINING PERIOD: 09 SEC
AS1
AS2
AS3
AS4

FIG. 11

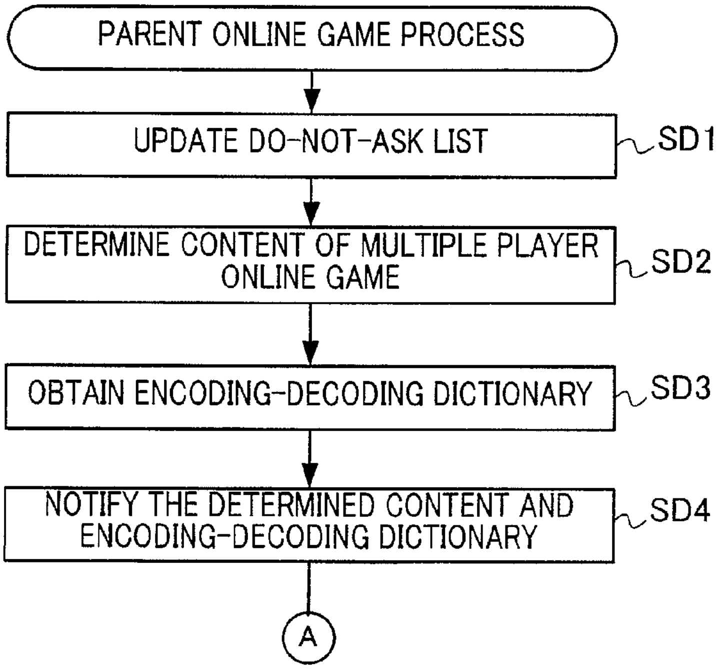

PARENT ONLINE GAME PROCESS
UPDATE DO-NOT-ASK LIST
SD1
DETERMINE CONTENT OF MULTIPLE PLAYER ONLINE GAME
SD2
OBTAIN ENCODING-DECODING DICTIONARY
SD3
NOTIFY THE DETERMINED CONTENT AND ENCODING-DECODING DICTIONARY
SD4
A

FIG. 12

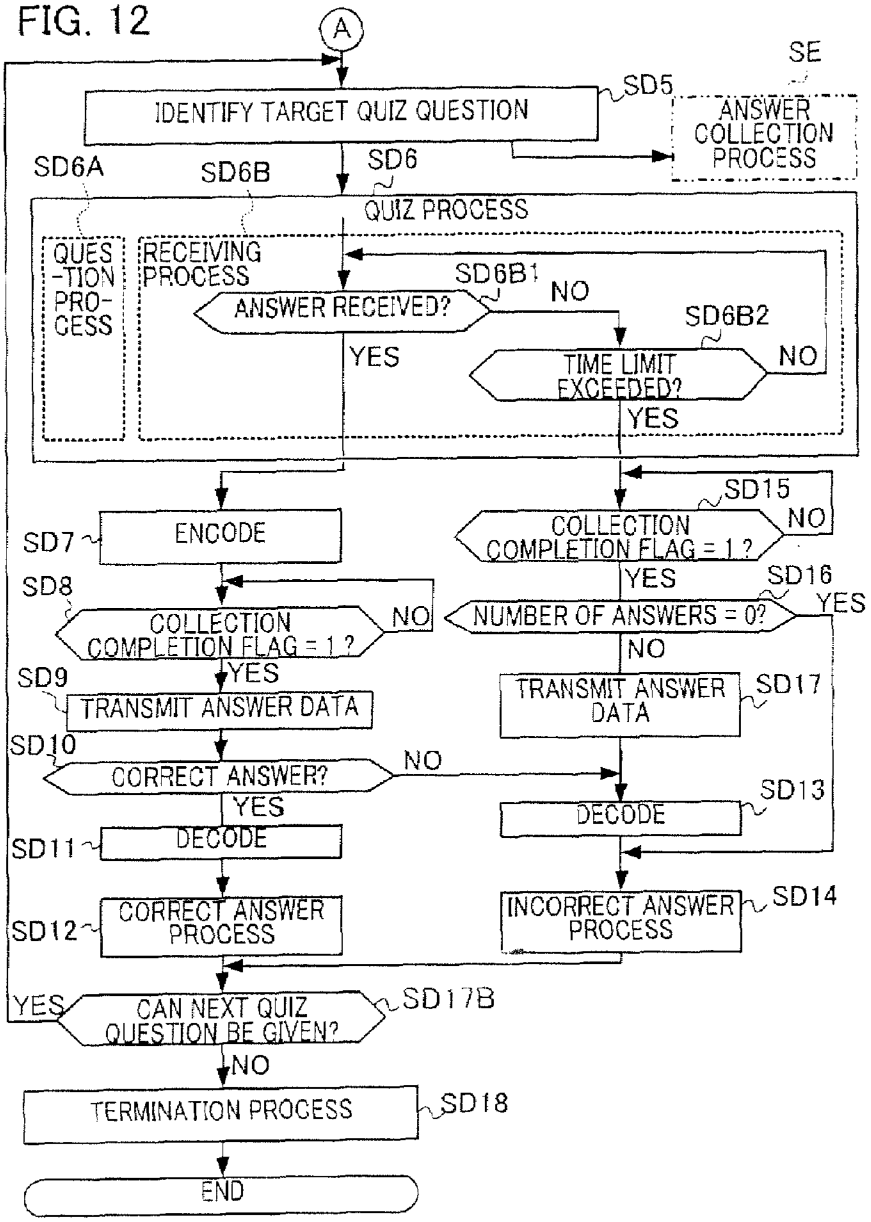
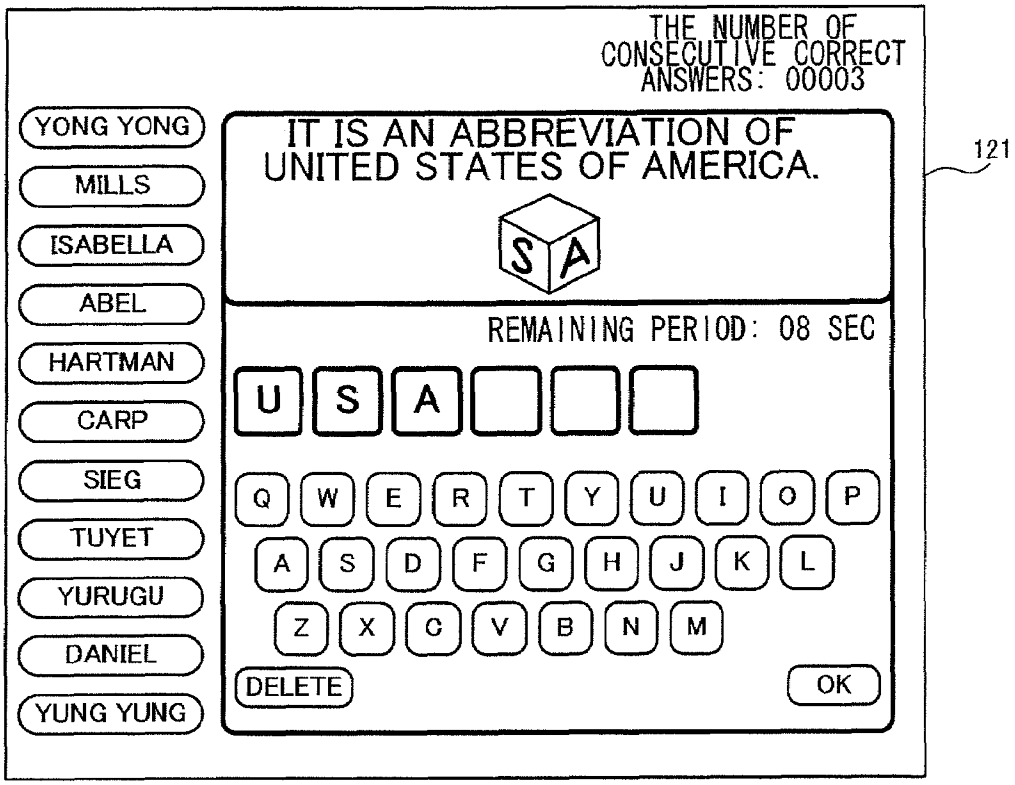

A
IDENTIFY TARGET QUIZ QUESTION
SD5
SE
ANSWER COLLECTION PROCESS
SD6A
SD6B
SD6
QUIZ PROCESS
QUES-TION PRO-CESS
RECEIVING PROCESS
SD6B1
ANSWER RECEIVED?
NO
SD6B2
YES
TIME LIMIT EXCEEDED?
NO
YES
SD7
ENCODE
SD15
COLLECTION COMPLETION FLAG = 1 ?
NO
YES
SD8
COLLECTION COMPLETION FLAG = 1 ?
NO
SD16
NUMBER OF ANSWERS = 0?
YES
NO
YES
SD9
TRANSMIT ANSWER DATA
TRANSMIT ANSWER DATA
SD17
SD10
CORRECT ANSWER?
NO
YES
DECODE
SD13
SD11
DECODE
SD12
CORRECT ANSWER PROCESS
INCORRECT ANSWER PROCESS
SD14
YES
CAN NEXT QUIZ QUESTION BE GIVEN?
SD17B
NO
TERMINATION PROCESS
SD18
END

FIG. 13



THE NUMBER OF CONSECUTIVE CORRECT ANSWERS: 00003

IT IS AN ABBREVIATION OF UNITED STATES OF AMERICA.

REMAINING PERIOD: 08 SEC

FIG. 14

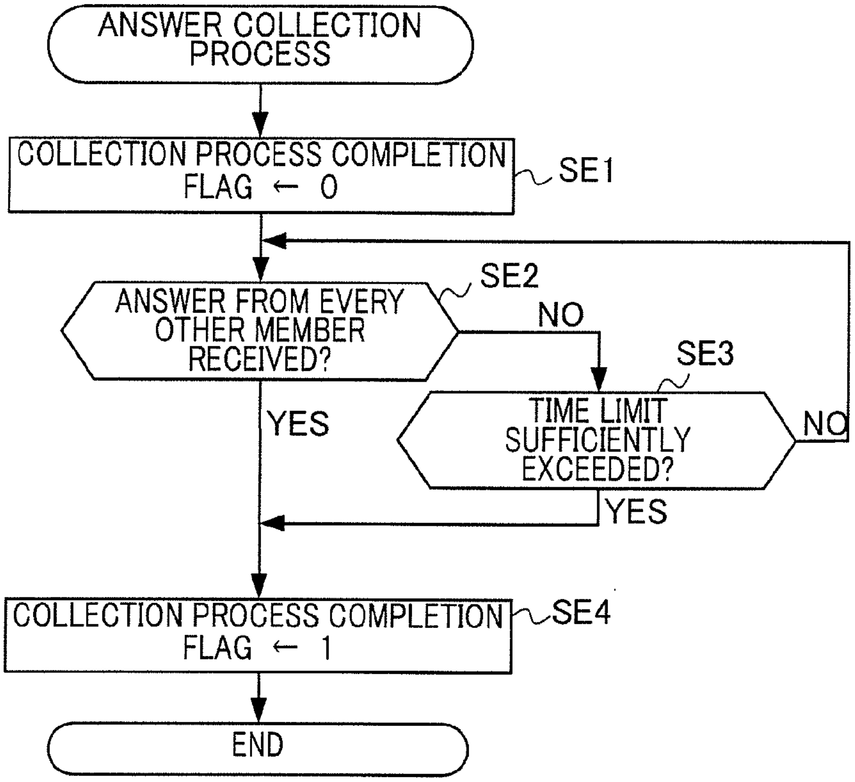
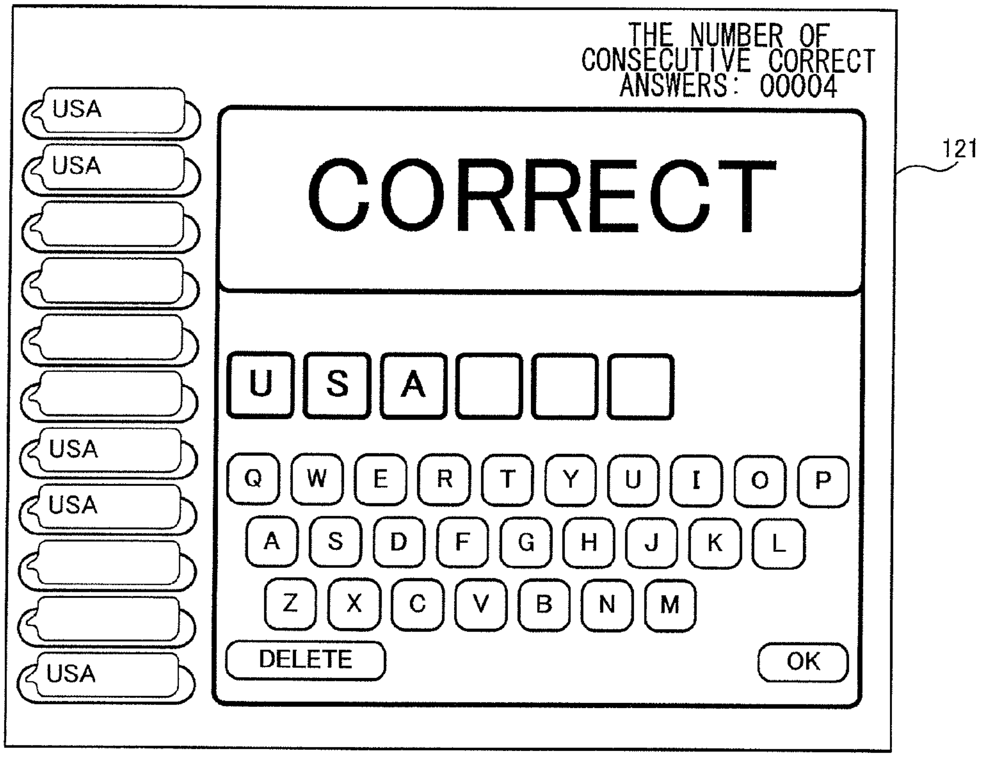

ANSWER COLLECTION PROCESS
COLLECTION PROCESS COMPLETION FLAG ← 0
SE1
SE2
ANSWER FROM EVERY OTHER MEMBER RECEIVED?
NO
SE3
TIME LIMIT SUFFICIENTLY EXCEEDED?
NO
YES
YES
COLLECTION PROCESS COMPLETION FLAG ← 1
SE4
END

FIG. 15



THE NUMBER OF CONSECUTIVE CORRECT ANSWERS: 00004

CORRECT

FIG. 16

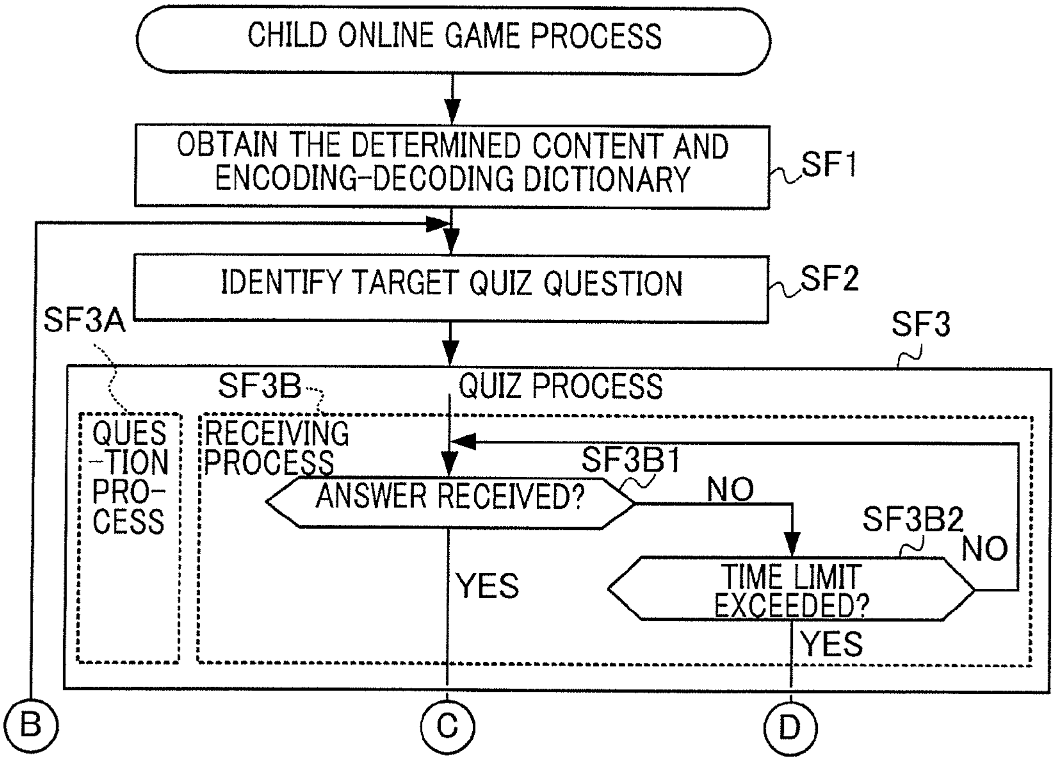

CHILD ONLINE GAME PROCESS
OBTAIN THE DETERMINED CONTENT AND ENCODING-DECODING DICTIONARY
SF1
IDENTIFY TARGET QUIZ QUESTION
SF2
SF3A
SF3
SF3B
QUIZ PROCESS
QUES-TION PRO-CESS
RECEIVING PROCESS
SF3B1
ANSWER RECEIVED?
NO
SF3B2
NO
YES
TIME LIMIT EXCEEDED?
YES
B
C
D

FIG. 17

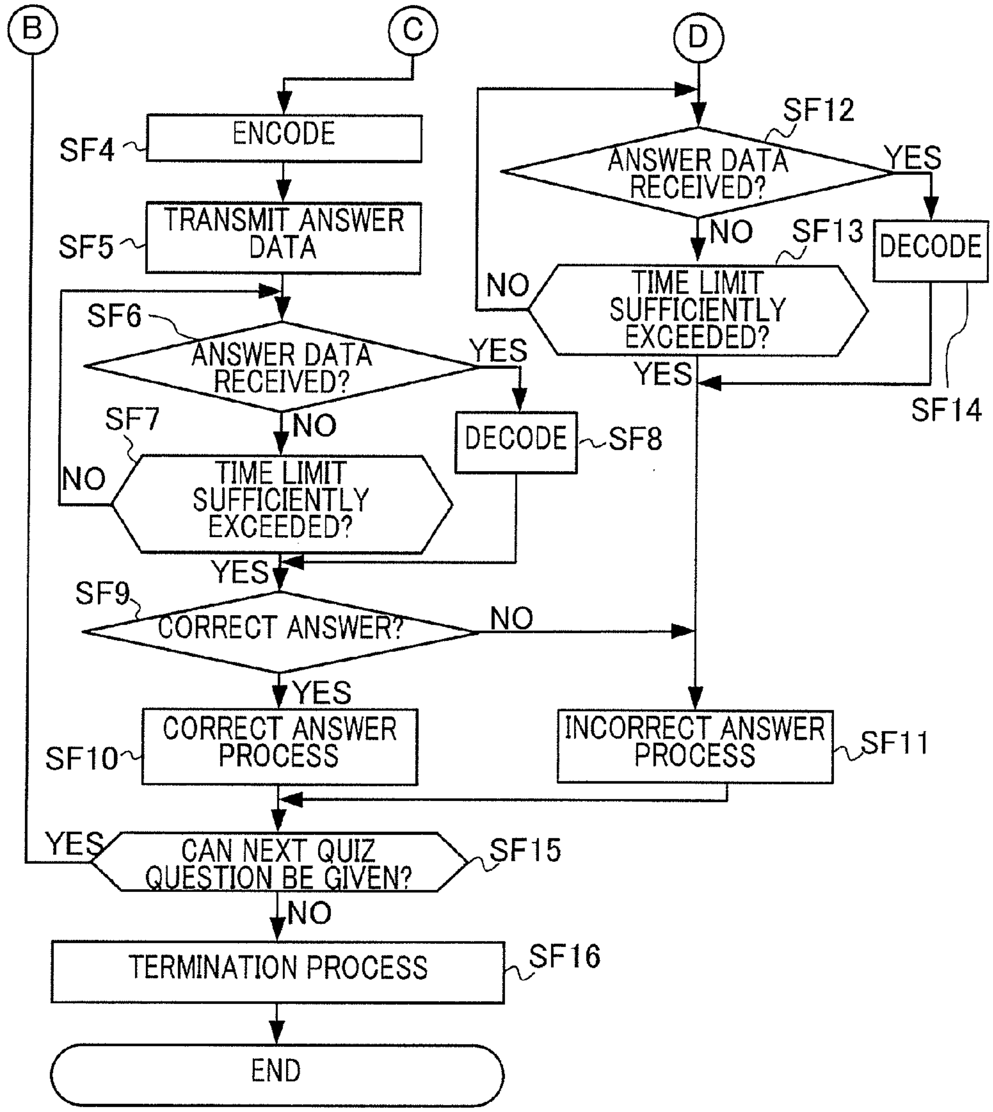

B
C
D
ENCODE
SF4
TRANSMIT ANSWER DATA
SF5
SF6
ANSWER DATA RECEIVED?
YES
NO
SF7
TIME LIMIT SUFFICIENTLY EXCEEDED?
NO
YES
DECODE
SF8
SF9
CORRECT ANSWER?
NO
YES
CORRECT ANSWER PROCESS
SF10
SF12
ANSWER DATA RECEIVED?
YES
NO
SF13
TIME LIMIT SUFFICIENTLY EXCEEDED?
NO
YES
DECODE
SF14
INCORRECT ANSWER PROCESS
SF11
YES
CAN NEXT QUIZ QUESTION BE GIVEN?
SF15
NO
TERMINATION PROCESS
SF16
END

FIG. 18

	ANSWER CHARACTER STRING	TO-BE-ENCODED BIT STRING	ENCODED BIT STRING
CORRECT	USA	011011000001	101
INCORRECT	USOA	0110110000110001	00101
INCORRECT	US	01101100	0111

FIG. 19

	ANSWER CHARACTER STRING	TO-BE-ENCODED BIT STRING	ENCODED BIT STRING
CORRECT	USA	011011000001	1010
INCORRECT	USOA	0110110000110001	001
INCORRECT	US	01101100	011

FIG. 20

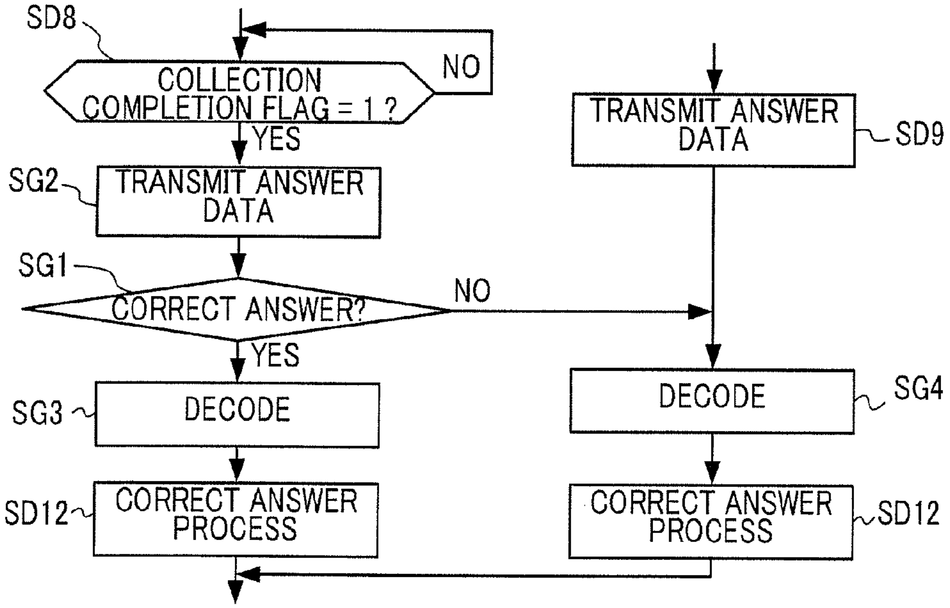
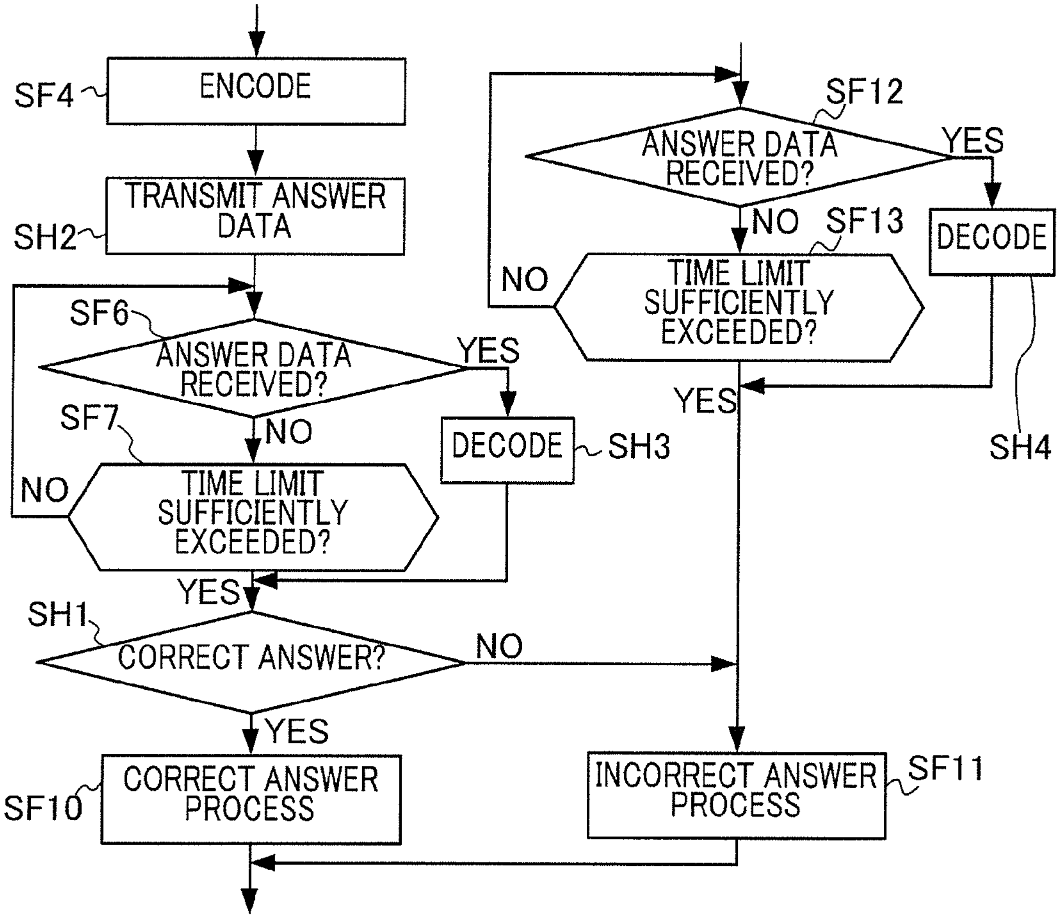


FIG. 21


SF4
ENCODE
SH2
TRANSMIT ANSWER DATA
SF6
ANSWER DATA RECEIVED?
YES
NO
SF7
TIME LIMIT SUFFICIENTLY EXCEEDED?
NO
YES
DECODE
SH3
SH1
CORRECT ANSWER?
NO
YES
SF10
CORRECT ANSWER PROCESS
SF12
ANSWER DATA RECEIVED?
YES
NO
SF13
TIME LIMIT SUFFICIENTLY EXCEEDED?
NO
YES
DECODE
SH4
INCORRECT ANSWER PROCESS
SF11

GAME TERMINAL, GAME SYSTEM THEREFOR, COMPUTER PROGRAM THEREFOR, AND COMPUTER READABLE RECORDING MEDIUM THEREWITH

TECHNICAL FIELD

[0001] The present invention relates to a game terminal that enables a player to play a quiz game including a quiz, a game system therefor, a computer program therefor, and also relates to a computer-readable recording medium therewith.

BACKGROUND ART

[0002] There is disclosed in Japanese Patent Application Laid-Open Publication No. 2006-055294 a game system in which a quiz game is advanced by repeatedly executing a series of processes for asking quiz questions to a player and receiving an answer for a quiz question from the player. In this game system, plural quiz questions are selected from a genre selected by a player, and the selected quiz questions are asked in a quiz game.

[0003] One type of quiz game is a multiple-player participation type online game in which plural players participate in a game simultaneously for playing the game. Because the same quiz question will be asked to the plural players in this type of quiz game, the quiz game will be more enjoyable, i.e., a player can enjoy a quiz game more, if each player is able to know not only the player's own answer, but also the answer of another player. In order to realize this, answers of each player for the same quiz question need to be shared among the plural game terminals. That is, data indicating an answer must be transmitted and received among game terminals.

[0004] Because there is a little flexibility in answers in a case in which the asked quiz question is of the multiple-choice type, the size of the data transmitted and received among game terminals can be sufficiently reduced so that the answers of each player can be shared. Therefore, in such a case, the amount of data transmitted and received among game terminals for sharing answers of each player can be minimized. In contrast, because there is a lot of flexibility in answers in a case in which the asked quiz question is of a type that requires a player to input a character string as an answer, it would be difficult to sufficiently reduce the data size of that transmitted and received among game terminals for sharing answers of each player. This means that in the conventional technique, it was difficult to minimize the size of the data, the data being transmitted and received among game terminals so that answers of each player can be shared in a case in which the asked quiz question is of a type that requires a player to input a character string as an answer.

DISCLOSURE OF INVENTION

[0005] The present invention has been achieved in view of the above-described circumstances, and it has as an object to provide a game terminal that can minimize the amount of data transmitted and received among game terminals while also enabling the sharing of answers of each player at plural game terminals in a multiple-player participation type online game in which a quiz question is asked in a manner in which a player is requested to input a character string as an answer, and to provide a game system therefor, a computer program therefor, and also to provide a computer-readable recording medium therewith.

[0006] In the following, description will be given of the present invention. It should be noted that reference numerals in the attached drawings are shown in parentheses to facilitate understanding of the present invention; however, this is not intended to limit the present invention to the embodiments as shown in the drawings.

[0007] The present invention provides a game terminal (**10**) for enabling a player to play a quiz game including a quiz, the game terminal comprising: a quiz processor (**150**, SD**6**, SD**10**, SD**12**, SD**14**, SF**3**, SF**9**, SF**10**, SF**11**) that, in the quiz, asks the player a quiz question, that receives an answer for the quiz question, and that informs a result of the quiz after the quiz is completed; a converter (encoder) (**150**, SD**7**, SF**4**, SG**2**, SH**2**) that converts (encodes) answer data representing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion (encoding) when the answer is correct; a transmitter (**150**, SD**9**, SF**5**, SG**2**, SH**2**) that transmits to another game terminal (**10**) the answer data converted by the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**); a receiver (**150**) that receives the answer data from another game terminal (**10**); and a reverse-converter (decoder) (**150**, SD**11**, SD**13**, SF**8**, SF**14**, SG**3**, SG**4**, SH**3**, SH**4**) that reverse-converts (decodes) the answer data received by the receiver (**150**). The present invention further provides a game system (**1**) that has a plurality of the above game terminals (**10**). The process of informing a result of a quiz indicates a process of determining whether answer data is correct and a correct answer process of notifying that the answer is correct by an image and by a sound or an incorrect answer process of notifying that the answer is incorrect by an image and by a sound.

[0008] In this game system (**1**), the size of transmitted and received data when a correct answer is received will be minimal (the shortest bit string), from among pieces of data transmitted and received among game terminals (**10**) so that answers of each player are shared among plural game terminals (**10**). Because in a quiz game, each player answers a quiz question, hoping to select the correct answer, it is possible according to the present invention to minimize the size of data transmitted and received among game terminals (**10**) while also enabling the sharing of answers of each player among game terminals (**10**) in a multiple-player participation type online game in which a quiz question is asked in a manner in which a player is requested to input a character string as an answer.

[0009] In the above game terminal (**10**), the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**) has an encoding-decoding dictionary storage device (**160**) that stores an encoding-decoding dictionary in which a to-be-encoded bit string and an encoded bit string are stored in a correlated manner for each of plural pieces of data that are different from one another; and an encoder (**150**, SD**7**, SF**4**) that encodes the answer data by using the encoding-decoding dictionary stored in the encoding-decoding dictionary storage device (**160**), and the answer data encoded by the encoder (**150**, SD**7**, SF**4**) is treated as the answer data converted by the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**).

[0010] According to a game system (**1**) having plural game terminals (**10**) of this embodiment, conversion and reverse conversion of answer data can be performed by simple processes.

[0011] In the above game terminal (**10**), the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**) has an encoding-decoding dictionary storage device (**160**) that stores an encoding-decoding dictionary in which a to-be-encoded bit string and an encoded bit string are stored in a correlated manner for each of plural pieces of data that are different from one another; an encoder (**150**, SD**7**, SF**4**) that encodes the answer data by using the encoding-decoding dictionary stored in the encoding-decoding dictionary storage device (**160**); and an extractor (**150**, SG**2**, SH**2**) that extracts a portion that is included in the bit string of the answer data encoded by the encoder (**150**, SD**7**, SF**4**) and that is not included in another encoded bit string, and the portion extracted by the extractor (**150**, SG**2**, SH**2**) may be treated as the answer data converted by the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**).

[0012] An example of the above portion is the head portion (for example, the first one bit) of an encoded bit string. In this embodiment, an encoding-decoding dictionary does not have to be dynamically reconfigured.

[0013] The above game system (**1**) has a server device (**40**) for communicating with each of the plural game terminals (**10**), and the server device (**40**) has a calculation mandatory data storage device (**430**) that stores, for each quiz question, data of parameters mandatory for accuracy rate calculation as calculation mandatory data; and a calculation mandatory data updater (**410**, SA**31**) that updates, based on the result of the quiz, the calculation mandatory data stored in the calculation mandatory data storage device (**430**), and the converter (**150**, SD**7**, SF**4**, SG**2**, SH**2**) may convert answer data showing the answer in such a manner that a bit string of the answer data after the conversion will be the shortest when the answer is correct, the conversion being performed in a case in which an accuracy rate of a quiz question calculated based on the calculation mandatory data stored in the calculation mandatory data storage device (**430**) is equal to or greater than a standard accuracy rate.

[0014] According to this embodiment, for a quiz question having an accuracy rate that is equal to or greater than the certain value, the size of transmitted and received data when a correct answer is received will be minimal (the shortest bit string), from among pieces of data transmitted and received among game terminals (**10**) so that answers of each player are shared among plural game terminals (**10**). Therefore, according to the present embodiment, the amount of data transmitted and received among game terminals (**10**) can be reliably reduced.

[0015] Furthermore, the present invention is a computer program (P**1**) or a computer program product (P**1**) for running on a computer (**10**) that enables a player to play a quiz game including a quiz, the computer program, when run on the computer (**10**) (or by a processor (**150**) of the computer), at least including computer-readable instructions for performing the steps of: processing a game, the processing including asking the player a quiz question, receiving an answer for the quiz question, and informing a result of the quiz after the quiz is completed (SD**6**, SD**10**, SD**12**, SD**14**, SF**3**, SF**9**, SF**10**, SF**11**); converting answer data showing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct (SD**7**, SF**4**, SG**2**, SH**2**); transmitting to another game terminal the answer data converted in the converting step (SD**7**, SF**4**, SG**2**, SH**2**) (**150**, SD**9**, SF**5**, SG**2**, SH**2**); receiving the answer data from another game terminal (**150**); and reverse-converting the answer data received in the receiving step (SD**11**, SD**13**, SF**8**, SF**14**, SG**3**, SG**4**, SH**3**, SH**4**). Preferably, the program (P**1**) or the computer program product (P**1**) may be downloaded to the computer (**10**) by a communication interface (**180**) by connecting to a network (**2**).

[0016] The present invention can be understood as a computer-readable recording medium that has recorded thereon the computer program (P**1**) or the computer program product (P**1**). Specifically, the present invention can be understood as that which is recorded, as data loadable into a memory of a computer, on a floppy (registered trademark) disk, a CD-ROM, or other types of computer-readable recording medium, for distribution.

[0017] The same effects as the game terminal (**10**) can be attained by a computer (**10**) that executes this computer program.

EFFECTS OF THE INVENTION

[0018] According to the present invention, it is possible to provide a game terminal which can curb the amount of data transmitted and received among game terminals while enabling the sharing of answers of each player at plural game terminals in a multiple-player participation type online game in which a quiz question is asked in a manner in which a player is requested to input a character string as an answer, a game system, a computer program therefor, and to provide a computer-readable recording medium therewith.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] FIG. **1** is a diagram illustrating a configuration of a game system **1** according to an embodiment of the present invention.

[0020] FIG. **2** is a diagram illustrating an external view of a game terminal **10** included in game system **1**.

[0021] FIG. **3** is a block diagram illustrating a configuration of game terminal **10**.

[0022] FIG. **4** is a diagram schematically illustrating a configuration of a quiz data set used at game terminal **10**.

[0023] FIG. **5** is a block diagram illustrating a configuration of a server device **40** included in game system **1**.

[0024] FIG. **6**A is a sequence chart illustrating an overview (the first half) of an operation performed in game system **1**.

[0025] FIG. **6**B is a sequence chart illustrating the overview (the second half) of the operation performed in game system **1**.

[0026] FIG. **7** is a flowchart illustrating an update process performed at server device **40**.

[0027] FIG. **8** is a flowchart illustrating a single game process performed at game terminal **10**.

[0028] FIG. **9** is a flowchart illustrating an association-based selection process included in the single game process.

[0029] FIG. **10** is a diagram illustrating an example of an image displayed at a screen **121** of game terminal **10**.

[0030] FIG. **11** is a flowchart illustrating a part (the first half) of a parent online game process performed at game terminal **10**.

[0031] FIG. **12** is a flowchart illustrating a part (the second half) of the parent online game process performed at game terminal **10**.

[0032] FIG. **13** is a diagram illustrating an example of an image displayed at screen **121**.

[0033] FIG. **14** is a flowchart illustrating an answer collection process included in the parent online game process.

[0034] FIG. **15** is a diagram illustrating an example of an image displayed at screen **121**.

[0035] FIG. **16** is a flowchart illustrating a part (the first half) of a child online game process performed at game terminal **10**.

[0036] FIG. **17** is a flowchart illustrating a part (the second half) of the child online game process performed at game terminal **10**.

[0037] FIG. **18** is a schematic diagram for describing a modification **1** of the present embodiment.

[0038] FIG. **19** is a schematic diagram for describing the modification **1**.

[0039] FIG. **20** is a diagram illustrating a partial flowchart of a parent online game process performed at a parent terminal in the modification **1**.

[0040] FIG. **21** is a diagram illustrating a partial flowchart of a child online game process performed at a child terminal in the modification **1**.

BEST MODE FOR CARRYING OUT THE INVENTION

[0041] In the following, description will be given of a preferred embodiment of the present invention with reference to the drawings. The specific configurations described below are merely examples, and the present invention encompasses various embodiments obtained by modifying the specific configurations within its scope.

1. Configuration

1-1. Game System

[0042] FIG. **1** is a block diagram illustrating a configuration of a game system **1** according to an embodiment of the present invention. Game system **1** is a system in which a player can play a quiz game including plural quiz questions. A quiz is a series of actions of a question master (game system **1** (specifically, a game terminal **10**, described later)) giving out a quiz question and an answerer (player) answering the question. In a quiz game, plural quizzes advance sequentially one by one. However, the embodiment may be modified such that plural quizzes advance in parallel in a quiz game.

[0043] A player can play a desired one of two types of quiz games using game system **1**. One type of the quiz game is a stand-alone game in which the number of participants is one. The other type of the quiz game is a multiple-player participation type online game (multiple-player online game) in which the number of participants is more than one (specifically, 12). The present embodiment may be modified such that the number of participants of the multiple player game is equal to or greater than 13 or equal to or less than 11.

[0044] A part of game system **1** is provided at game facilities F (F**1**, F**2**, F**3**, . . .) such as a venue at which a player who visits the venue can play a game. Specifically, there are provided at each game facility F, a LAN (Local Area Network) **25**, one or plural game terminals (computer) **10** connected to LAN **25**, enabling a player to play a quiz game, and a router **20** that connects LAN **25** with the Internet **2**.

[0045] Game terminal **10** is used by a player and is a terminal device (arcade game device) by use of which a player can play a quiz game. The game terminal performs a game process including a quiz process of giving a player a quiz question and receiving an answer to the quiz question from the player. The game process is a process for causing a player to play a quiz game.

[0046] Router **20** has a function of building, on the Internet **2**, a VPN (Virtual Private Network) which is a virtual dedicated network. Game terminal **10** is capable of communicating, via LAN **25**, with another game terminal **10** in game facility F where this game terminal **10** is located, and is also capable of communicating, via LAN **25**, router **20**, and the Internet **2**, with game terminal **10** at another game facility F.

[0047] Game system **1** has a server device (computer) **40** that provides various services (processes) (described below) to every game terminal **10** and a router **30** that connects server device **40** to the Internet **2**. Server device **40** is capable of communicating with each game terminal **10** via router **30** and the Internet **2**. In other words, server device **40** and all the game terminals **10** can communicate with one another. Server device **40** is a single unit, but the present embodiment may be modified such that server device **40** is formed of plural units.

1-2. Game Terminal

[0048] FIG. **2** is a diagram illustrating an external view of game terminal **10** included in game system **1**. A player of game terminal **10**, in front of a body **110** of game terminal **10**, can see an image displayed on a screen **121** of a display unit (specifically, a monitor) **120** provided on body **110** and can also hear sounds emitted from a speaker **131** provided on body **110**, and can touch a transparent touch panel **141** provided on body **110** and covering screen **121**.

[0049] In a quiz, a player is expected to understand a quiz question given by game terminal **10** based on an image displayed on screen **121** and on sounds emitted from speaker **131** and to answer the quiz question by touching touch panel **141**.

[0050] FIG. **3** is a block diagram illustrating a configuration of game terminal **10**. As shown in the figure, game terminal **10** has a processor **150** that performs various processes, described later, an operation unit **140** that is operated by a player and that supplies processor **150** with operation data corresponding to the operation of the player, a display unit **120** that displays an image on screen **121** by using image data supplied from processor **150**, a sound emitter **130** that emits sounds from speaker **131** by using sound data supplied from processor **150**, a terminal storage device **160** that stores various types of data, and a communication interface **180**.

[0051] Operation unit **140** is provided with touch panel **141**. Touch panel **141** supplies processor **150** with operation data corresponding to a position touched by a player. Processor **150** is, for example, one or plural CPUs (Central Processing Units) and performs a game process described below. Communication interface **180** is for transmitting a signal to, and receiving a signal from, LAN **25**, and relays data between processor **150** and LAN **25**. Processor **150** uses communication interface **180** to transmit data to, and to receive data from, another device. The present embodiment may be modified so as to use a video projector as display unit **120**. In this case, a screen on which an image is projected will be screen **121**, and operation unit **140** will be provided with a keyboard or buttons, etc.

[0052] A memory area of terminal storage device **160** is divided into a volatile area RT1 for which power is required for retaining the memory content and a non-volatile area RT2 for which power is not required. Non-volatile area RT2 is further divided into a non-rewritable area RT**21** in which the memory content is non-rewritable, and a rewritable area RT**22** in which the memory content is rewritable. Volatile area RT1 can be made, for example, by a RAM (Random Access Memory), non-rewritable area RT**21**, for example, by a ROM (Read Only Memory), a rewritable area RT**22**, for example, by a hard disk.

[0053] Reserved in rewritable area RT**22** is a terminal quiz table QT**1**. In terminal quiz table QT**1**, there are stored, for each quiz question, quiz question ID data indicating a quiz question ID (identifier) for uniquely identifying a quiz question, a quiz data set for performing a quiz using a quiz question, genre data representing a genre of the quiz question, question style data representing a question style of the quiz question. Quiz question IDs, genres and question styles are the same in game system **1**. That is, it is ensured that a quiz question having the same quiz question ID is the same quiz question, that the genre of the same quiz question is the same, and that the question style of the same quiz question is the same, among devices in game system **1**.

[0054] A genre (or a category) is a field of knowledge from which a question is given to a player and is, for example, a field of sports or a field of learning, etc. A question style is, for example, a multiple choice question for answering a quiz question by selecting one of plural answer options, a type-in type of question for answering a quiz question by entering characters, etc. The genre and the question style of each quiz question are fixed. The present embodiment may be modified such that plural genres or plural question styles are associated with one quiz question.

[0055] FIG. **4** is a schematic diagram illustrating a configuration of a quiz data set used in game terminal **10**. The quiz data set includes quiz question data representing a quiz question, correct answer data representing a correct answer of the quiz question, and keyword group data representing a keyword group including plural keywords (character strings) that are related to a quiz element. In a case in which the question style is of the multiple-choice type, a quiz question includes plural answer options, and the quiz question data includes data representing plural answer options. Only one of these plural answer options is the correct answer.

[0056] A quiz element is an element essential to a quiz question and specifically is a quiz question or a correct answer. In a case in which the question style is a multiple-choice question, incorrect answers will also be a quiz element. There are plural types of keyword groups: namely, a quiz-question related keyword group including keywords that are related to a quiz question, a correct-answer related keyword group including keywords that are related to a correct answer, and an incorrect-answer related keyword group including keywords that are related to incorrect answers. Thus, different types of keyword groups include keywords related to quiz elements that differ from group to group. The incorrect-answer related keyword group is provided only in a case in which the question style is of the multiple-choice type.

[0057] A quiz data set includes quiz-question related keyword group data representing a quiz-question related keyword group and correct-answer related keyword group data representing a correct-answer related keyword group. A quiz data set of a quiz question for which the question style is of the multiple choice type further includes incorrect-answer related keyword group data representing an incorrect-answer related keyword group. Furthermore, each keyword group data includes data representing one common keyword. That is, each keyword group data includes at least one keyword that is in common to all quiz data sets and also to all keyword groups.

[0058] Stored in the volatile area RT**1** are a terminal-side-do-not-ask list BL**1** for prohibiting asking of inadequate questions to be asked and consecutive correct answer count data C representing the number of consecutive correct answers. The inadequate questions to be asked include a quiz question that is inadequate for asking such as a quiz question for which a correct answer cannot be given. A terminal-side-do-not-ask list BL**1** is a list containing pieces of quiz question ID data of inadequate questions to be asked and is delivered from server device **40**. As described below, plural quiz questions are asked sequentially one by one in a quiz game. The number of consecutive correct answers is initially set to 0, and it increments by one every time a correct answer is given to a quiz question and is reset to 0 when an incorrect answer is given. In volatile area RT**1**, a collection process completion flag FL (described below) and an encoding-decoding dictionary table DT (described below) can be retained. There are stored in encoding-decoding dictionary table DT, for each quiz question, quiz question ID data and encoding-decoding dictionary. The encoding-decoding dictionary is data for encoding and decoding answer data indicating an answer of a player.

[0059] Stored in the non-rewritable area RT**21** is a computer program P**1**. Computer program P**1** includes computer-readable instructions, and when computer program P**1** is run and executed by processor **150**, computer program P**1** causes game terminal **10** to perform a game process. The game process includes a terminal-side grouping process for forming a group with another game terminal **10**, a single game process for causing a player to play a stand-alone type game, and an online game process for causing a player to play a multiple player online game.

[0060] A group is a collection of game terminals **10**, and one multiple-player online game advances by the same game terminals **10** in the same group communicating with one another. The number of groups to which one game terminal **10** belongs is one. In each group, from among game terminals **10** belonging to each group, one terminal is a parent (master) game terminal **10** (parent terminal) capable of directly transmitting data to, and receiving data from, every other game terminal **10**, and each of the other terminals is a child (client) game terminal **10** (child terminal) capable of directly transmitting data only to, and receiving data only from, the parent terminal (**10**). In the following description, in each group, each game terminal **10** that belongs to the group will be called a "member" of the group.

1-3. Server Device

[0061] FIG. **5** is a block diagram illustrating a configuration of server device **40** included in game system **1**. As shown in the figure, server device **40** has a processor **410** that performs various processes, described later, a communication interface **420** that relays data between processor **410** and router **30**, and a server storage device **430** that stores various data. Processor **410** is, for example, one or plural CPUs and uses communication interface **420** to transmit data to, and receive data from, each game terminal **10**.

[0062] The memory area of server storage device **430** is divided into a volatile area RT**3** for which power is required for retaining the memory content and a non-volatile area RT**4** for which power is not required. Volatile area RT**3** can be made, for example, by a RAM and stores a server-side-do-not-ask list BL**2**. This server-side-do-not-ask list BL**2** is read from volatile area RT**3** and is transmitted as terminal-side-do-not-ask list BL**1** to each game terminal **10**. It is to be noted that the present embodiment may be modified such that server-side-do-not-ask list BL**2** is stored not in volatile area

RT3 but in non-volatile area RT4. Except, the memory area for server-side-do-not-ask list BL2 must be a rewritable area.

[0063] Non-volatile area RT4 is, for example, a ROM or a hard disk. Reserved in non-volatile area RT4 is a server-side quiz table QT2 and an adequacy determination table JT. The server-side quiz table QT2 stores, for each quiz question, a quiz question ID, difficulty level data representing a difficulty level (difficulty degree) of a quiz question, genre data, question style data, accuracy rate data representing an accuracy rate (percentage of correct answers) of a quiz question, quiz question ask number data representing the number of times a quiz question is asked, correct answer number data representing the number of correct answers of a quiz question, and encoding-decoding dictionary data representing an encoding-decoding dictionary.

[0064] The ask number of a quiz question is the number of times the quiz question is asked in all the multiple-player online games played in game system **1**. The number of correct answers of a quiz question is the number of times players gave a correct answer for the quiz question in multiple-player online games in the entire game system **1**. The accuracy rate of each quiz question is variable and is updated in accordance with a game result. The accuracy rate of each quiz question is calculated by dividing the number of correct answers of this quiz question by the number of times this quiz question is asked. The accuracy rate of a quiz question for which the number of times the questions is asked is 0 is determined based on the difficulty level of the quiz question.

[0065] As is clear from the foregoing description, the quiz question ask number data and the correct answer number data are accuracy rate calculation mandatory data necessary for calculation of the accuracy rate (hereinafter, this will be referred to simply as "calculation mandatory data"). The number of times a quiz question is asked and the number of times a correct answer is received for the quiz question are parameters mandatory for accuracy rate calculation.

[0066] The encoding-decoding dictionary is a dictionary for encoding and decoding data in transmitting and receiving the data among members. In each encoding-decoding dictionary, for each of plural pieces of data that are different from one another, a to-be-encoded bit string of each piece and an encoded bit string are associated with each other. In each encoding-decoding dictionary, plural to-be-encoded bit strings are different from one another, and plural encoded bit strings are different from one another. Each encoding-decoding dictionary is initially configured in such a manner that, when the accuracy rate of a corresponding quiz question is equal to or greater than a predetermined standard accuracy rate, the encoded bit string for correct answer data indicating a correct answer for the quiz question is the shortest among all encoded bit strings in the encoding-decoding dictionary, and otherwise configured in a random manner. It is to be noted that the present embodiment can be modified in such a way that quiz data sets are stored in server-side quiz table QT2 so that the server-side quiz table serves as a superset of terminal-side quiz table QT1.

[0067] When a quiz data set in server-side quiz table QT2 is manually updated, the quiz data set in terminal-side quiz table QT1 is also updated in conjunction. This data synchronization is carried out by transmitting the quiz data set from server device **10** to each game terminal **10**. In a case in which there are numerous game terminals **10**, it would take a considerable amount of time for the updating. Therefore, the updating should be completed during hours when game facility F is closed.

[0068] The adequacy determination table JT has stored therein, for each difficulty level (difficulty degree) of a quiz question, adequate range data representing a range (adequate range) of the accuracy rate that is adequate as the accuracy rate of a quiz question of a certain difficulty level. The adequate range of the accuracy rate is a range within which an adequate accuracy rate falls and an inadequate accuracy rate does not fall, and is predetermined. The adequate range data is for identifying a quiz question that has an accuracy rate that does not fall within the adequate range of the accuracy rate, and it can be data indicating the upper limit and the lower limit of the adequate range or can be data indicating only the lower limit of the adequate range. Because 100% is the upper limit of the accuracy rate, the data indicating only the lower limit of the adequate range is also data indicating an adequate range. It is to be noted that, because the adequate range data also shows an inadequate range of accuracy rate as the accuracy rate of a quiz question of a corresponding difficulty level, the adequate range data can also be regarded as data for identifying a quiz question that has an accuracy rate that falls within the inadequate range of the accuracy rate.

[0069] Non-volatile area RT4 stores a computer program P2. Computer program P2 includes computer-readable instructions, and when the computer program P2 is executed by processor **410**, causes server device **40** to perform a server process. The server process includes a server-side grouping process for grouping plural game terminals **10**, an update process for updating data stored in server storage device **430**, a list delivery process for delivering terminal-side-do-not-ask list BL1 to game terminal **10**, and a dictionary delivery process for delivering encoding-decoding dictionary data to game terminal **10**. These processes can be executed in parallel.

2. Operation

2-1. Overview

[0070] FIGS. **6**A and **6**B are sequence charts illustrating an operation of game system **1**. FIG. **6**B shows an operation that is continued from each operation of each game terminal **10** and server device **40** shown in FIG. **6**A. As shown in FIGS. **6**A and **6**B, while game system **1** is operating, the server-side grouping process is continued at server device **40**. On the other hand, as shown in FIG. **6**A, at game terminal **10**, a process corresponding to a type of quiz game which a player will be playing (hereinafter, this will be referred to as a "play object"). In a case in which a play object is a multiple-player online game, processor **150** of game terminal **10** first performs a terminal-side grouping process.

[0071] In the terminal-side grouping process, processor **150** first uses display unit **120** and operation unit **140**, thereby to provide a UI (User Interface) for causing a player to input a play name as a name of a player for use in the multiple-player online game. The UI can be freely selected and may be, for example, a virtual keyboard. It is to be noted that the present embodiment can be modified such that a player selects one of many play names prepared in advance. Furthermore, a player may be able to use a player name that was input or selected in the past by using game terminal **10** or another device.

[0072] In the terminal-side grouping process, processor **150** first transmits, to server device **40**, a grouping request for requesting grouping. On the other hand, processor **410** of server device **40** is continuously executing a server-side grouping process, and in a case in which a grouping request is received from game terminal **10**, a grouping is performed for deciding game terminal **10** as the parent population. Specifically, processor **410** transmits an empty grouping response to game terminal **10** which will be a parent terminal of each group and transmits a grouping response to game terminal **10** which will be a child terminal of each group, with the grouping response including data of a communication address of game terminal **10** which will be a parent terminal of the same group. In an example shown in FIG. **6**A, after a grouping request is transmitted to server device **40** from plural game terminals **10** which will belong to the same group, a grouping response is transmitted from server device **40** to these plural game terminals **10**, but it is to be noted that this is merely an example.

[0073] In the terminal-side grouping process, processor **150**, having received a grouping response including data representing a communication address, transmits, to game terminal **10** (parent terminal) of this communication address, play name data representing a play name input to game terminal itself (child terminal). By this process, game terminal **10**, to which the play name data is transmitted, can obtain a communication address of the transmitter of the play name data. As a result, data can be transmitted and received among freely selected game terminals **10** that will belong to each group directly or indirectly via game terminal **10** that will be a parent terminal of the same group.

[0074] In the terminal-side grouping process, processor **150** of game terminal **10** that has received the play name data writes a set of the play name data and data representing the communication address of the transmitter into volatile area RT**1**, to transmit this set to every game terminal **10** except for game terminal **10** that has transmitted data of the same play name, from among game terminals **10** of which a communication address processor **150** has obtained. Game terminal **10** having received the set writes this data into volatile area RT**1**. A group is thus formed. From among data contained in the transmitted set, data representing a communication address of the transmitter of play name data is used at the receiver of this data only for identifying the play name data. That is, the data representing the communication address is only necessary to be able to identify the play name data at the receiver side of the set. However, in a case in which data other than the data representing the communication address of a transmitter of the play name data is used, game terminal **10** that will be a parent terminal needs to know the correspondence between this other data and the data representing the communication address.

[0075] When a group is formed, a terminal-side group process is completed at each member of the group. The process relating to a play name is necessary because the play names of all the members are displayed on every screen **121** of each member of the same group. Therefore, if the present embodiment is modified so that such a display is not performed, the process associated with a play name is no longer necessary. In this case, freely selected data, instead of data representing a play name, will be transmitted from game terminal **10**, which will be a child terminal to game terminal **10** that will be a parent terminal.

[0076] In a case in which the play object is a multiple-player online game, as shown in FIG. **6**B, processor **150** of game terminal **10** that has finished the terminal-side grouping process performs an online game process (a parent online game process or a child online game process) corresponding to the attribute (parent or child) of its own. The details of each online game process will be described later, but just an overview is given now.

[0077] In the parent online game process, processor **150** of a parent terminal first transmits, to server device **40**, a list delivery request for requesting delivery of terminal-side-do-not-ask list BL**1**. Processor **410** of server device **40**, every time it receives the list delivery request, performs a list delivery process. That is, processor **410** reads server-side-do-not-ask list BL**2** from volatile area RT**3**, for transmission as return as terminal-side-do-not-ask list BL**1** to the transmitter of the list delivery request. Processor **150** of a parent terminal receives the transmitted terminal-side-do-not-ask list BL**1** to overwrite data in volatile area RT**1** with the received list and determines the details of a quiz game (a multiple-player online game in this case) based on the terminal-side-do-not-ask list BL**1**. Specifically, processor **150** selects the regulated number of the candidates of quiz questions to be given in a quiz game (hereinafter referred to as “candidate questions”) and determines the order of asking these candidate questions. The regulated number is a plural number equal to or greater than the maximum number of quiz questions asked in a quiz game and is predetermined.

[0078] In the parent online game process, processor **150** of the parent terminal first transmits to server device **40** a dictionary delivery request for requesting delivery of encoding-decoding dictionary data representing an encoding-decoding dictionary of the selected, prescribed candidate questions. The dictionary delivery request contains pieces of quiz question ID data of the selected, prescribed candidate questions, with these pieces of quiz question ID data being listed in the decided order of asking. Processor **410** of server device **40**, every time it receives a dictionary delivery request, performs a dictionary delivery process. Specifically, processor **410** reads from server-side quiz table QT**2** pieces of encoding-decoding dictionary data corresponding to the regulated number of pieces of quiz question ID data contained in the dictionary delivery request, generates a dictionary delivery response in which the pieces of encoding-decoding dictionary data are listed in the order of asking, and transmits the response to the transmitter of the dictionary delivery request. Processor **150** of a parent terminal receives the transmitted dictionary delivery response and stores, in encoding-decoding dictionary table DT, the pieces of encoding-decoding dictionary data in the dictionary delivery response in association with corresponding quiz question ID data.

[0079] On the other hand, in the child online game process, processor **150** of a child terminal first shares, with the parent terminal, the selection of plural quiz questions, the order of asking the plural quiz questions, and the encoding-decoding dictionary in a parent terminal belonging to the same group. When this sharing is completed at every child terminal of the same group, a multiple-player online game of the determined content is performed in this group.

[0080] When the multiple-player online game is finished, processors **150** of all the member game terminals **10** of the group in which the game was played each transmits to server device **40** play result data representing a result of the play of the game, to terminate the online game process. The play result data is data for each player. The play result data includes, for each and every quiz question included in a multiple-player online game that has ended, quiz result data indicating a result of a quiz (hereinafter, this will be referred to as a "quiz result").

[0081] Processor **410** of server device **40**, every time play result data is received, performs an update process. Specifically, processor **410** updates data stored in server storage device **430** in accordance with the received play result data. As a result, data such as server-side quiz table QT**2** and server-side-do-not-ask list BL**2** is updated. It is to be noted that the present embodiment can be modified so that a parent terminal of a group collects pieces of play result data of all the game terminals **10**, for transmission to server device **40**.

[0082] On the other hand, in a case in which the play object is a stand-alone type game, processor **150** of game terminal **10** performs a single game process. The details of the single game process will be described below, and just an overview is given here. In the single game process, processor **150** of game terminal **10** transmits the above-described list delivery request to server device **40**. As a result, terminal-side-do-not-ask list BL**1** will be transmitted from server device **40** to this game terminal **10** as a reply. Processor **150** of this game terminal **10** receives terminal-side-do-not-ask list BL**1** and overwrites data in volatile area RT**1** with the received list, to determine the content of a quiz game (a stand-alone type game in this case) based on terminal-side-do-not-ask list BL**1**. The stand-alone type game is performed at game terminal **10** under the determined content. When the stand-alone type game is completed, processor **150** of game terminal **10** ends a single game process.

2-2. Update Process

[0083] FIG. **7** is a flowchart illustrating an update process performed at server device **40**. In the update process, quiz result data included in play result data received from game terminal **10** is selected sequentially one by one, and server-side quiz table QT**2** is, or server-side quiz table QT**2** and server-side-do-not-ask list BL**2** are, updated based on the information of the selected quiz result data. Processor **410** of server device **40** first determines whether there is any quiz result data that is not yet selected for which an update process is not yet performed, from among plural pieces of quiz result data contained in the result data received from game terminal **10** (SA**1**). In a case in which a result of the determination is negative, processor **410** ends the update process.

[0084] In a case in which a result of the determination in Step SA**1** is affirmative, processor **410** first selects, from among the plural pieces of quiz result data, a piece of quiz result data that has not yet been selected (SA**2**). Subsequently, processor **410** performs, based on the selected piece of quiz result data, an update process of server-side quiz table QT**2** (SA**3**). The details of this update process depend on an accuracy rate, as will be described below.

[0085] In the update process of server-side quiz table QT**2**, processor **410** first updates calculation mandatory data based on the selected piece of quiz result data (SA**31**). The quiz result data is data indicating a result of a quiz question, and specifically indicates a quiz question ID of the asked quiz question and whether an answer to the question was correct.

[0086] In Step SA**31**, processor **410** updates quiz question ask number data and correct answer number data corresponding to a quiz question ID in the selected piece of quiz result data (hereinafter referred to as "selected quiz ID") or quiz question ask number data corresponding to the selected quiz ID so that the number of correct answers for a quiz question having the selected quiz ID increases by 1 only when the number of times of asking the quiz question having the selected quiz ID increases by 1 and when the correct or incorrect data in the selected piece indicates that a correct answer was given, i.e., when the quiz result data indicates that an answer to the quiz question was correct.

[0087] Subsequently, processor **410** refers to server-side quiz table QT**2** to calculate an accuracy rate of a quiz question having the selected quiz ID and updates accuracy rate data so that accuracy rate data indicating the accuracy rate of a quiz question of the selected quiz ID represents the calculated accuracy rate (SA**32**). Subsequently, processor **410** determines whether the accuracy rate of a quiz question having the selected quiz ID changes to a value equal to or greater than the predetermined standard accuracy rate as a result of the update performed in Step SA**32** (SA**33**). This result of the determination becomes positive only when the accuracy rate of a quiz question having the selected quiz ID is equal to or greater than the standard accuracy rate and when the past accuracy rate of the quiz question having the selected quiz ID (an accuracy rate represented by accuracy rate data immediately before the update of this time) is less than the standard accuracy rate.

[0088] In a case in which a result of the determination in Step SA**33** is affirmative, processor **410** performs a configuration process in which correct answer data is prioritized as a process of reconfiguring the encoding-decoding dictionary (SA**34**). Specifically, processor **410** updates the encoding-decoding dictionary of a quiz question of the selected quiz ID in such a manner that a bit string of encoding correct data indicating a correct answer of a quiz question having the selected quiz ID will be the shortest among all encoded bit strings in the encoding-decoding dictionary. The update process of server-side quiz table QT**2** is thus completed. It is to be noted that the present invention may be modified so that the configuration process of Step SA**34** is always performed if the accuracy rate of a quiz question having the selected quiz ID is equal to or exceeds the standard accuracy rate.

[0089] In a case in which a result of the determination in Step SA**33** is negative, processor **410** determines whether the accuracy rate of a quiz question having the selected quiz ID has changed to below the predetermined standard accuracy rate as a result of the update in Step SA**32** (SA**35**). This result of determination will be affirmative only when the accuracy rate of a quiz question having the selected quiz ID is less than the standard accuracy rate and when the past accuracy rate of the quiz question having the selected quiz ID (an accuracy rate represented by accuracy rate data immediately before the update of this time) is equal to or greater than the standard accuracy rate. In a case in which a result of the determination is affirmative, processor **410** performs, as a process of reconfiguring the encoding-decoding dictionary, a configuration process in which correct answer data is not prioritized (SA**36**). The content of the reconfiguration process may be freely determined. For example, the correspondence between a to-be-encoded bit string and an encoded bit string may be determined at random. Specifically, processor **410** reconfigures the encoding-decoding dictionary of the quiz question having the selected quiz ID so that the freely determined length of encoded data corresponds to to-be-encoded data of correct answer data corresponding to the selected quiz ID. The update process of this server-side quiz table QT**2** is thus completed. On the other hand, a result of the determination in

Step SA**35** is negative, i.e., in a case in which there is no change between the accuracy rate before update and after update, relative to the standard accuracy rate, processor **410** ends the update process of server-side quiz table QT**2**.

[0090] When the update process of server-side quiz table QT**2** is completed, processor **410** determines, based on the adequacy determination table JT, whether the accuracy rate of a quiz question having the selected quiz ID is not within the adequate range corresponding to the difficulty level of this quiz question (SA**4**). In a case in which a result of the determination is affirmative, processor **410** updates server-side-do-not-ask list BL**2** (SA**5**). Specifically, processor **410** updates server-side-do-not-ask list BL**2** so as to include quiz question ID data indicating the selected quiz ID. The process then returns to Step SA**1**. In a case in which a result of the determination Step in SA**4** is negative, the process returns to Step SA**1**.

[0091] Thus, in the update process, server-side quiz table QT**2** is updated based on play result data received from game terminal **10**, and server-side-do-not-ask list BL**2** is updated so as to include quiz question ID data indicating a quiz question ID of a quiz question for which the accuracy rate is not within the adequate range. It is to be noted that the present embodiment can be modified so that quiz question ID data indicating a quiz question ID of a quiz question for which the accuracy rate is within the adequate range is deleted from server-side-do-not-ask list BL**2**.

2-3. Single Game Process

[0092] FIG. **8** is a flowchart illustrating a single game process performed at game terminal **10**. In the single game process, processor **150** of game terminal **10** updates terminal-side-do-not-ask list BL**1** (SB**1**). Specifically, processor **150** transmits a list delivery request to server device **40**, receives, from server device **40**, terminal-side-do-not-ask list BL**1**, and overwrites data in volatile area RT**1** with the received terminal-side-do-not-ask list BL**1**.

[0093] Processor **150** then, based on terminal-side-do-not-ask list BL**1**, determines the content of a stand-alone type game (SB**2**). Specifically, processor **150** provides a UI for causing a player to specify a genre and a question style, and in a case in which the genre and the question style are specified, processor **150** defines all the questions of the specified genre and question style as a population for selecting candidates (hereinafter referred to as the "population for candidate selection"), selects from this population for candidate selection the regulated number of quiz questions identified by quiz question IDs that are not included in terminal-side-do-not-ask list BL**1** as candidate questions, and determines the order of asking the selected, regulated number of candidate questions. This selection and determination are performed by using, for example, random numbers.

[0094] In a case in which the question style is not specified, the question style is automatically determined, and the population for candidate selection is defined based on the determined question style. In this case, the question style is determined by using, for example, random numbers. In a case in which a genre is not specified, all the quiz questions in the specified or determined question style will be the population for candidate selection in the above selection, and the selection of candidate questions from this population for candidate selection is performed by the association-based selection process for selecting the regulated number of candidate questions that have an association that is different from a genre.

[0095] FIG. **9** is a flowchart illustrating an association-based selection process included in the single game process. In the association-based selection process, processor **150** of game terminal **10** first focuses on one quiz question that is not yet focused from the population for candidate selection (SC**1**). Specifically, processor **150** focuses on one piece of quiz question ID data that is not yet focused on from among pieces of quiz question ID data of terminal-side quiz table QT**1**. This one piece of quiz question ID data is determined by using, for example, random numbers. Subsequently, processor **150** selects, based on the specified or determined question style, a type of keyword group (SC**2**). For example, in a case in which the specified or determined question style is multiple-choice, processor **150** selects one of a quiz-question related keyword group, a correct-answer related keyword group, or an incorrect-answer related keyword group. The present embodiment may be modified so that plural types are selected from these keyword groups.

[0096] Subsequently, processor **150** specifies a search keyword (SC**3**). This specification is performed by defining, based on keyword group data in terminal-side quiz table QT**1**, a keyword group that is the same as the type of keyword group selected in Step SC**2** as a population (hereinafter referred to as the "population for keyword selection") from among keyword groups of a quiz question that is being focused on and by selecting from this population one keyword that has not yet been selected. This selection is performed by using, for example, random numbers. It is to be noted that the present embodiment may be modified such that plural search keywords are identified.

[0097] Subsequently, processor **150** determines whether the regulated number of quiz questions can be selected as candidate questions based on terminal-side quiz table QT**1**, terminal-side-do-not-ask list BL**1**, the search keyword, and the selected type of keyword group (SC**4**). Specifically, it is determined whether the population for candidate selection includes quiz questions equal to or greater than the regulated number, with the quiz question being a quiz question of the specified or determined question style, with the selected type of keyword group of the quiz question including the search keyword, and with the quiz question ID of the quiz question not being included in terminal-side-do-not-ask list BL**1** (hereinafter referred to as "quiz questions that can become candidate").

[0098] In a case in which a result of the determination in Step SC**4** is negative, the process returns to Step SC**3**. That is, the process of Steps SC**3** to SC**4** is repeated until a result of the determination in Step SC**4** is positive. Since each of all the keyword groups of all quiz data sets includes at least one keyword that is common to all the keyword groups, a result of the determination in Step SC**4** will eventually change to be positive. In a case in which a result of the determination in Step SC**4** is positive, processor **150** selects, as candidate questions, each of the regulated number of quiz questions that can become candidate (SC**5**), to end the association-based selection process.

[0099] The present embodiment may be modified so that, in a case in which the specified or the determined question style is multiple-choice, the sum of correct-answer related keyword group and incorrect-answer related keyword group is defined as a population for keyword selection. That is, the sum of plural types of keyword groups can be defined as the population for keyword selection. The present embodiment may be modified so that the keyword group data is prepared for each of the answer options.

[0100] As shown in FIG. **8**, once the content of the standalone type game is determined, processor **150** then identifies a target quiz question as a quiz question to be asked in the next quiz process (SB**3**). Specifically, from among candidate questions that have not yet been asked, a candidate question that is first in the order of asking is regarded as a target quiz question. Processor **150** then executes a quiz process (SB**4**). The quiz process is a process of performing a quiz. In the quiz process, processor **150** performs a question process for asking a target quiz question by using display unit **120** (SB**4**A) and performs a receiving process of receiving answers from a player by using display unit **120** and operation unit **140** (SB**4**B). The question process and the receiving process are performed by using a quiz data set (quiz question data) for performing a quiz using quiz questions to be asked. An example of an image displayed on screen **121** is shown in FIG. **10**.

[0101] FIG. **10** is a diagram illustrating an example of an image displayed on screen **121** of game terminal **10**. This example illustrates a case in which the specified or determined quiz question is of the multiple-choice type. Displayed in the upper display area of screen **121** in the diagram is a character string of the target quiz question, "Which is the image of Tokyo Station?", and displayed in the lower display area of screen **121** in the diagram are images of plural answer options AS**1** to AS**4**. A player touches an area on screen **121** that is occupied by the image of a desired answer option, thereby selecting the answer option, i.e., to answer the quiz question to be asked.

[0102] In the receiving process, processor **150** determines whether an answer from a player is received (SB**4**B**1**), and in a case in which a result of the determination is negative, processor **150** determines whether the length of time that has elapsed from the start of the receiving process has exceeded the time limit (SB**4**B**2**). In a case in which a result of the determination is negative, processor **150** returns the process to Step S**4**B**1**. In short, until an answer of a player is received (until a player gives an answer), or until the elapsed time exceeds the time limit, processor **150** waits for an answer from the player.

[0103] The time limit is a predetermined certain length of time for limiting a period for receiving an answer (selectable period) and is, for example, 60 seconds. An image displayed on screen **121** includes an image representing the remaining period until the elapsed time exceeds the time limit ("09 seconds" in FIG. **10**). It is to be noted that in the present embodiment, the start of the receiving process is the same as the start of the quiz process, but the receiving process may be started after the quiz process is started.

[0104] Processor **150**, in a case in which a result of the determination in Step SB**4**B**1** or Step SB**4**B**2** of the receiving process is affirmative, terminates the quiz process to perform the next process. Specifically, in the case of terminating the quiz process after a result of the determination in Step SB**4**B**2** is positive, i.e., in the case of terminating the quiz process by the elapsed time exceeding the time limit, processor **150** performs an incorrect answer process (SB**5**). Specifically, processor **150** performs a process of displaying on screen **121** an image for informing a player that the answer is incorrect, a process for emitting from speaker **131** of sound emitter **130** a sound for informing the player that the answer is incorrect, and a process of initializing the number of consecutive correct answers.

[0105] On the other hand, in a case in which the quiz process is terminated by a result of the determination in Step SB**4**B**1** becoming affirmative, i.e., in case in which the quiz process is terminated by an answer being received, processor **150** determines whether the received answer is correct based on the received answer and correct answer data contained in a quiz data set for performing a quiz in which a target quiz question is used (SB**6**). In a case in which a result of the determination is negative, the process proceeds to Step SB**5**. In a case in which a result of the determination is affirmative, processor **150** performs a correct answer process (SB**7**). Specifically, processor **150** performs a process for displaying on screen **121** an image informing a player that the answer is correct, a process for emitting from speaker **131** of sound emitter **130** a sound for informing that the answer is correct and a process of updating consecutive correct answer count data C so that the number of consecutive correct answers increases by 1.

[0106] Processor **150**, having completed the correct answer process or the incorrect answer process, determines whether the next quiz question can be given (SB**8**). The content of this determination should be determined as appropriate in accordance with the specifications of a quiz game. For example, in a case in which a quiz game is a game terminated after the quiz process is performed for a certain number of times, processor **150** can determine whether the number of completed quiz processes reaches a certain number of times. As another example, in a case in which a quiz game is a game that is terminated within a certain length of time, processor **150** can determine whether the length of time that has elapsed from the start of a quiz game is equal to or exceeds the time limit.

[0107] In a case in which a result of the determination in Step SB**8** is positive, the process returns to Step SB**3**. In other words, the process of Steps SB**3** to SB**8** is repeated until a result of the determination in Step SB**8** turns negative. It is to be noted that, since the regulated number is a number that is equal to or greater than the maximum number of quiz questions given in a quiz game, quiz questions to be asked will not be in short supply. On the other hand, in a case in which a result of the determination in Step SB**8** is negative, processor **150** uses display unit **120** to cause a result of the play to be displayed on screen **121** (SB**9**), to terminate the single game process.

2-4. Parent Online Game Process

[0108] FIGS. **11** and **12** are diagrams each illustrating a part of a flowchart of a parent online game process performed at game terminal **10**. In the parent online game process, processor **150** of a parent terminal updates terminal-side-do-not-ask list BL**1** (SD**1**). This process is the same as Step SB**1** in FIG. **8**.

[0109] Subsequently, processor **150**, based on terminal-side-do-not-ask list BL**1**, determines the content of a multiple-player online game (SD**2**). This process is the same as Step SB**2**, except that in Step SD**2** processor **150** does not provide a UI for causing a player to specify a genre or a question style. Instead, processor **150** automatically determines the genre and the question style, or only the question style.

[0110] Subsequently, processor **150** obtains the encoding-decoding dictionary (SD**3**). Specifically, processor **150** first initializes the encoding-decoding dictionary table DT and then transmits, to server device **40**, a dictionary delivery request for requesting a dictionary which has the content in accordance with the regulated number of candidate questions selected in Step SD**2**. Subsequently, processor **150** receives a dictionary delivery response which is transmitted from server device **40** as a reply, to store, in the encoding-decoding dictionary table DT, the regulated number of pieces of encoding-decoding dictionary data in the received dictionary delivery response and the regulated number of pieces of quiz question ID data in the transmitted dictionary delivery request in a correlated manner, one to one, based on the listed position of the questions IDs. It is to be noted that initializing the encoding-decoding dictionary table DT means deleting all pieces of data from encoding-decoding dictionary table DT.

[0111] Subsequently, processor **150** notifies, to every other member, the details of the determination in Step SD**2** and the encoding-decoding dictionary obtained in Step SD**3** (SD**4**). Specifically, processor **150** transmits, to every other member, the pieces of quiz question ID data of the selected, regulated number of candidate questions, data indicating the order of asking these candidate questions, and pieces of encoding-decoding dictionary data representing encoding-decoding dictionaries of these candidate questions. Processor **150** then identifies a target quiz question as a quiz question to be asked in the subsequent quiz process (SD**5**). The multiple-player online game is started before Step SD**5** at the parent terminal.

[0112] Processor **150** then performs the quiz process (SD**6**) and an answer collection process of collecting answers received by other members (SE). These processes are performed in parallel with each other. The quiz process in Step SD**6** is the same as the quiz process in Step SB**4**. An example of an image displayed on screen **121** in this quiz process is illustrated in FIG. **13**.

[0113] FIG. **13** is a diagram illustrating an example of an image displayed on screen **121** of game terminal **10**. In this example, the determined question style is of the type-in style, in which a target quiz question is displayed on a display area of the upper portion of the figure in screen **121**, and a virtual keyboard is displayed on a display area of the lower portion of the figure in screen **121**. The target quiz question includes character strings, "It is an abbreviation of 'The United States of America'." and an image of a rotating cube. Each surface of this cube has a character-by-character depiction of every character contained in "USA", which is the correct answer. A player looks at the character strings of the target quiz question and the image to think of the character string of an answer, inputs the character string of the answer by inputting the characters one by one by touching an area occupied by a desired key of the virtual keyboard, and touches an area occupied by a predetermined key ("OK" key in the figure), thus answering the quiz question to be asked. The character string of the input answer is displayed on a display area of a nearly central portion of the figure on screen **121**. Furthermore, the play names of players who are using the members other than this game terminal are displayed in an area of the left portion of the figure in screen **121**. It is to be noted that the Delete key in the figure is a key for deleting one character input most recently from among the characters in the input character string.

[0114] FIG. **14** is a flowchart illustrating an answer collection process included in the parent online game process. The answer collection process commences simultaneously with a process corresponding to Step SB**4**B**2** (SD**6**B**2**). In the answer collection process, processor **150** first reserves, in volatile area RT**1**, a collection process completion flag FL indicating whether a process of collecting answers received at other members has been completed and sets the value as **0** (not yet completed) (SE**1**).

[0115] Subsequently, processor **150** determines whether it has received answer data representing an answer from every other member (SE**2**), and in a case in which a result of the determination is negative, processor **150** determines whether the length of time that has elapsed from the start of the receiving process has sufficiently exceeded the time limit (SE**3**). In a case in which a result of the determination is negative, processor **150** returns the process to Step SE**2**. That is, processor **150** waits for answer data from other members until it receives answer data from every other member or the elapsed time has sufficiently exceeded the time limit.

[0116] Once a result of the determination in Step SE**2** or SE**3** turns affirmative, processor **150** changes the value of collection process completion flag FL in volatile area RT**1** to 1 (completed) (SE**4**), to end the answer collection process. In the present embodiment, the result of the determination in Step SE**3** will not become affirmative only by the elapsed time exceeding the time limit, so as to prevent the failure to collect answer data due to process delay and communication delay.

[0117] In the quiz process (SD**6**) in FIG. **12**, processor **150**, in a case in which a result of the determination in a process (SD**6**B**1**) corresponding to Step SB**4**B**1** turns positive, or a result of the determination in a process (SD**6**B**2**) corresponding to Step SB**4**B**2** turns positive, terminates the quiz process to perform the next process. Specifically, processor **150**, in the case of terminating the quiz process by receiving an answer (SD**6**B**1**; YES), refers to the encoding-decoding dictionary of the target quiz question, to encode answer data representing this answer (SD**7**). The length of a bit string of the encoded answer data (to be more precise, the encoding-decoding dictionary of a quiz question to be asked) will be the shortest when an accuracy rate of a quiz question to be asked is equal to or greater than a standard accuracy rate and when the answer data is the same as correction data indicating a correct answer of the quiz question to be asked. That is, the answer data is converted so that a bit string after conversion will be the shortest when the answer data indicates a correct answer.

[0118] Subsequently, processor **150** determines whether the collection process completion flag FL is 1 (completed) (SD**8**). In a case in which a result of the determination is negative, the process returns to Step SD**8**. In other words, processor **150** waits for a process of collecting answers received at other members to be completed. Processor **150** then, to every other member, transmits its own answer data and collected pieces of answer data (SD**9**). Every piece of answer data has been encoded. It is to be noted that the present embodiment may be modified so that transmission of answer data is omitted in a case in which the destination of the answer data is a member from which the answer data was collected.

[0119] Subsequently, processor **150** determines whether, based on the answer received at this game terminal and on correct answer data included in the quiz data set for performing a quiz using the target quiz question, the answer is correct (SD**10**). In a case in which a result of the determination is positive, processor **150** refers to the encoding-decoding dictionary of the target quiz question, to decode each collected piece of answer data (SD**11**). This decoding is reverse-conversion (i.e., decoding) of encoding (conversion) in Step SD**7** and Step SF**4** (described later).

[0120] Processor **150** then performs a correct answer process that is the same as that of Step SB**7** (SD**12**), except that in this correct answer process, processor **150** uses display unit **120** to display on screen **121** not only the answer received by this game terminal, but also answers received by other members. FIG. **15** illustrates an example of an image displayed on screen **121** in this process.

[0121] FIG. **15** is a diagram illustrating an example of an image displayed on screen **121** of game terminal **10**. This example illustrates a case in which the determined question style is of the type-in style. As it is clear when compared with FIG. **13**, the answers of players who are using the member terminals except for this game terminal are displayed in an area in the left portion of the figure in screen **121** in FIG. **15**. These answers are displayed in a manner in which it shows which answer is the answer of which player.

[0122] On the other hand, in a case in which a result of the determination in Step SD**10** is negative, processor **150** refers to the encoding-decoding dictionary of the target quiz question, to decode each collected piece of answer data (SD**13**), and performs an incorrect answer process that is the same as Step SB**5** (SD**14**). However, in this incorrect answer process, processor **150** uses display unit **120** to display in screen **121** not only the answer received by this game terminal but also the answers received by other members. These answers are displayed in a manner showing which answer is the answer of which player.

[0123] On the other hand, in a case in which the quiz process was terminated by a result of the determination in Step SD**6**B**2** turning positive, i.e., the quiz process was terminated by the elapsed time exceeding the time limit, processor **150** first determines whether the collection process completion flag FL is 1 (completed) (SD**15**). In a case in which a result of the determination is negative, the process returns to Step SD**15**. In other words, processor **150** waits until a process of collecting answers received by the other members is completed. Subsequently, processor **150** determines whether the number of collected answers (answer data) is 0 (SD**16**). A result of this determination turns affirmative only in a case in which no answer was received by any of the members.

[0124] In a case in which a result of the determination in Step SD**16** is negative, processor **150** transmits, to every other member, its own answer data and the collected answer data (SD**17**). This process is the same as the process in Step SD**9**. The process then advances to Step SD**13**. On the other hand, in a case in which a result of the determination in Step SD**16** is affirmative, the process advances to Step SD**14**. That is, in a case in which an answer was not received at any of the members, transmission of answer data (S**17**) and decoding of answer data (S**13**) are skipped.

[0125] Processor **150**, having finished a correct answer process or an incorrect answer process, determines whether the next quiz question can be given (SD**17**), and in a case in which a result of the determination is positive, the process returns to Step SD**5**. That is, until a result of the determination in Step SB**17** turns negative, the process from Steps SD**5** to SD**17** (and the answer collection process (SE)) is repeated. On the other hand, in a case in which a result of the determination in Step SB**8** is positive, processor **150** performs a termination process of terminating the parent online game process (SD**18**), to end the parent online game process. A multiple-player online game ends before the termination process at the parent terminal.

[0126] In this termination process, processor **150** uses display unit **120** to display a result of the play on screen **121** and transmits, to server device **40**, play result data representing a result of the play of a player who is using this game terminal with respect to a multiple-player online game, which has ended.

[0127] In generating this play result data, processor **150** generates pieces of quiz result data for each of quiz questions included in the multiple-player online game, which has ended. From among these pieces of quiz result data, a piece of quiz result data indicating that the answer was correct is always a piece of quiz result data for a quiz question for which a result of determination in Step SD**10** became positive.

2-5. Child Online Game Process

[0128] FIGS. **16** and **17** are diagrams each illustrating a part of a flowchart of a child online game process performed at game terminal **10**. In the child online game process, processor **150** of a child terminal first obtains, from a parent terminal of the same group, the content of a multiple-player online game and the encoding-decoding dictionary (SF**1**). Specifically, pieces of quiz question ID data of the regulated number of candidate questions selected at the member parent terminal, data representing the order of asking these candidate questions, and encoding-decoding dictionary data representing the encoding-decoding dictionary of these candidate questions are received from the parent terminal.

[0129] Subsequently, processor **150** identifies a target quiz question which is a quiz question to be given in the next quiz process (SF**2**). A multiple-player online game starts at a child terminal before Step SF**2**. Processor **150** then performs a quiz process (SF**3**). The quiz process in Step SF**3** is the same as the quiz process in Step SB**4**. In Step SF**3**, an image that is the same as the image shown in FIG. **13** is displayed on screen **121**. In the quiz process in Step SF**3**, processor **150** terminates the quiz process in a case in which a result of the determination in a process (SF**3**B**1**) corresponding to Step SB**4**B**1** or a result of the determination in a process (SF**3**B**2**) corresponding to Step SB**4**B**2** becomes affirmative, to perform the next process.

[0130] In a case of terminating the quiz process by receiving an answer, processor **150** refers to the encoding-decoding dictionary of the target quiz question, to encode answer data representing the answer (SD**4**). Subsequently, processor **150** transmits, to the member parent terminal, the encoded answer data of this game terminal (SD**5**). Processor **150** then determines whether it has received answer data from the member parent terminal (SF**6**), and in a case in which a result of the determination is negative, processor **150** determines whether the length of time that has elapsed from the start of the receiving process in Step SF**3**B has sufficiently exceeded the time limit (SF**7**). In a case in which a result of this determination is negative, the process returns to Step SF**6**.

[0131] Processor **150**, in a case in which a result of the determination in Step SF**6** turns affirmative by receiving answer data from the member parent terminal, refers to the encoding-decoding dictionary of the target quiz question, to decode every received answer data (SF**8**). Processor **150** then determines whether its own answer is correct based on its own answer data and correct answer data contained in the quiz data set for performing a quiz using the target quiz question (SF**9**). On the other hand, in a case in which a result of the determination in Step SF**7** turns positive by the elapsed time having sufficiently exceeded the time limit, the process advances to Step SF**9**.

[0132] In a case in which a result of the determination in Step SF**9** turns positive, processor **150** performs a correct answer process that is the same as Step SD**12** in FIG. **12** (SF**10**). On the other hand, in a case in which a result of the determination in Step SF**9** turns negative, processor **150** performs an incorrect answer process that is the same as Step SD**14** in FIG. **12** (SF**11**). Meanwhile, in the case of terminating the quiz process by the elapsed time exceeding the time limit, processor **150** performs the same process as Steps SF**6** to SF**8** (SF**12** to SF**14**). When this process is completed, the routine advances to Step SF**11**.

[0133] Processor **150**, having completed the correct answer process or the incorrect answer process, determines whether the next quiz question can be given (SF**15**), and in a case in which a result of the determination is affirmative, the process returns to Step SF**2**. In other words, the process from Steps SF**2** to SF**15** is repeated until a result of the determination in Step SF**15** becomes negative. On the other hand, in a case in which a result of the determination in Step SF**15** is affirmative, processor **150** performs the same termination process (SF**16**) as the above-described terminal process (SD**18**), to terminate the child online game process. In the termination process of Step SF**16**, from among pieces of quiz result data included in the multiple-player online game, which has ended, a piece of quiz result data indicating that an answer was correct is always a piece of quiz result data for a quiz for which a result of the determination in Step SF**9** became positive.

[0134] It is to be noted that the multiple-player online game ends at a child terminal before the termination process.

3. Effects

[0135] In the following, description will be given of the effects that can be attained by the present embodiment. It is to be noted that, if any of the effects that will be described in the following is not necessary, an element other than an element for bringing about a necessary result may be removed from game system **1**.

[0136] In the present embodiment, game terminal **10** has processor **150**, and processor **150** performs a processing, in a quiz, of asking a player a quiz question, receiving an answer for the quiz question, and informing the result of the quiz after the quiz is completed (SD**6**, SD**10**, SD**12**, SD**14**, SF**3**, SF**9**, SF**10**, SF**11**), a process of converting answer data representing the received answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct (SD**7**, SF**4**), transmitting to another game terminal **10** the converted answer data (SD**9**, SF**5**), and a process of receiving answer data from another game terminal **10**, and a process of reverse-converting the received answer data (SD**11**, SD**13**, SF**8**, SF**14**).

[0137] Therefore, in game system **1**, the size of transmitted and received correct answer data when a correct answer is received will be minimal (the shortest bit string), from among pieces of correct answer data transmitted and received among members so that answers of each player are shared among members. Because, in a quiz game, each player answers a quiz question, with the hope of determining the correct answer, it is possible, according to the present invention, to minimize the size of data transmitted and received among members while enabling the sharing of answers of each player among members in a multiple-player participation type online game in which a quiz question is asked in a manner in which a player is requested to input a character string as an answer.

[0138] Furthermore, according to the present embodiment, conversion and reverse-conversion of answer data are carried out by encoding and decoding by using an encoding-decoding dictionary. That is, according to the present embodiment, conversion and reverse-conversion of answer data can be performed by a simple process.

[0139] Furthermore, according to the present embodiment, game system **1** has server device **40**, and server device **40** has processor **410** and server storage device **430**. Server storage device **430** stores therein accuracy rate calculation mandatory data that is mandatory for accuracy rate calculation of a quiz question. Processor **410** updates accuracy rate calculation mandatory data stored in server storage device **430** based on a result of a quiz at each of plural game terminals **10**. Processor **150** of game terminal **10** converts answer data representing the received answer in such a manner that the converted bit will be the shortest when the answer is correct, in a case in which an accuracy rate of a quiz question calculated using the updated accuracy rate calculation mandatory data is equal to or greater than a standard accuracy rate. In short, the conversion and reverse-conversion are performed for a quiz question having an accuracy rate that is equal to or exceeds the certain value. Therefore, the amount of answer data transmitted and received among members can be reliably reduced.

[0140] In the present embodiment, game terminal **10** has processor **150** and terminal storage device **160**, and there are stored in terminal storage device **160**, for each quiz question, keyword group data representing a keyword group including plural keywords, quiz question data representing the quiz question, correct answer data representing a correct answer of the quiz question, and genre data indicating a genre of the quiz question. Processor **150**, by using computer program **P1** stored in terminal storage device **160**, performs a process (SB**4**B, SD**6**B) of receiving an answer from a player in each quiz, a process (SB**6**, SD**10**) of determining whether an answer received in each quiz is correct by using correct answer data representing a correct answer of a quiz question given in the quiz, a process (SC**3**) of dynamically specifying one search keyword, a process (SC**5**) of selecting plural quiz questions having association with one another by selecting plural quiz questions for each of which keyword group data representing a keyword group that includes the search keyword is stored in terminal storage device **160**, and a process (SB**4**A, SD**6**A) of giving the selected plural quiz questions in one quiz game, by using quiz question data representing each quiz question.

[0141] Thus, at game terminal **10**, one search keyword is dynamically specified in each quiz game, and plural quiz questions that are associated by the specified search keyword are given to a player. Therefore, according to game terminal **10**, the associations among plural quiz questions that are given to a player can be used as a game element, so that the game-worthiness of a quiz game can be increased.

[0142] Furthermore, at game terminal **10**, because neither quiz question data or correct answer data is concurrently used as keyword group data, the freedom in data format of quiz question data or correct answer data does not have to be reduced. For example, data representing an image can be quiz question data.

[0143] Furthermore, in the present embodiment, there are plural types of keyword groups corresponding to each quiz question, and the plural types of keyword groups each contains a keyword that is related to a quiz element that is different from group to group. Processor **150** of game terminal **10** uses computer program P1 stored in terminal storage device **160**, to perform a process (SC**2**) of selecting one of the plural types. Also, quiz questions selected in a process (SC**5**) of selecting plural quiz questions are limited to a quiz question that includes the specified search keyword and to a quiz question for which keyword group data representing the selected type of keyword group is stored in terminal storage device **160**. Therefore, by use of game terminal **10**, a question as to which quiz element is focused on in associating the plural quiz questions can also be a game element, and the game-worthiness of a quiz game can be further enhanced.

[0144] Additionally, in the present embodiment, there is stored in terminal storage device **160**, for each quiz question, genre data indicating a genre of a quiz question separately from keyword group data, and processor **150** of game terminal **10** performs a process (SB**2**, SD**2**) of specifying one genre, a process (SB**2**, SD**2**) of selecting plural quiz questions for which genre data representing the identified genre is stored in terminal storage device **160**, and a process (SB**4**A, SD**6**A) of giving a player the selected plural quiz questions in one quiz game, by using quiz question data representing each quiz question. Therefore, according to game terminal **10**, in each quiz game, plural quiz questions that are associated with one another can be given to a player, the association relating to something other than a genre, and moreover, the plural quiz questions that are selected from the specified one genre can be selected.

4. Modifications

[0145] The above embodiment may be modified as described in the following.

4-1. Modification for Encoding-Decoding Dictionary

4-1-1. Modification 1

[0146] For example, as illustrated in FIG. **18**, the encoding-decoding dictionary may be configured such that, for each quiz question, only the first bit of an encoded bit string corresponding to answer data (answer character string) which is the same as the correct answer data (correct answer) has the particular value ("1" in the figure) from among the first bit of bit strings of all pieces of encoded data of encoding-decoding dictionary. In a case in which the encoding-decoding dictionary is thus configured, the encoding-decoding dictionary may be configured such that, from among encoded bit strings contained in each encoding-decoding dictionary data, an encoded bit string that is different from an encoded bit string corresponding to a to-be-encoded data which is the same as the correct answer data is the shortest, as shown in FIG. **19**.

[0147] By configuring the encoding-decoding dictionary as shown in FIG. **18** or FIG. **19**, it is possible to provide a game system in which only the first bit of encoded answer data has to be transmitted in a case in which answer data to be exchanged among members is correct answer data. FIG. **20** illustrates a part of a flowchart of the parent online game process performed by a parent terminal in this mode; and FIG. **21** shows a part of a flowchart of the child online game process performed by a child terminal in this mode. Those parts that are not illustrated in FIG. **20** or **21** are the same as those in FIGS. **12** and **17**.

[0148] In this mode, only the first bit of encoded answer data has to be referred to in order to determine whether an answer is correct (Step SG**1** in FIG. **20**, Step SH**1** in FIG. **21**). Furthermore, in this mode, in a case in which answer data representing an answer received by a child terminal is correct answer data, the child terminal transmits, to the member parent terminal, only the first one bit of encoded answer data instead of transmitting the encoded answer data itself (Step SH**2** in FIG. **21**), and correct answer data can be obtained at the parent terminal by decoding the one bit (Step SG**3** in FIG. **20** and Step SG**4** in FIG. **20**). The same can be said of answer data representing an answer received at a parent terminal (Step SG**2** in FIG. **20**, Step SH**3** in FIG. **21**, Step SH**4** in FIG. **21**).

[0149] Thus, a method of converting answer data in a manner in which the bit string of the answer data will be the shortest after the conversion when the answer data indicates a correct answer is not limited to a method according to the above-described embodiment. For example, as described above, a game terminal may extract a portion that is included in the encoded bit string of the answer data indicating a correct answer and that is not included in another encoded bit string, so that the portion extracted by the extractor is treated as the converted answer data. In this example, an encoding-decoding dictionary does not have to be dynamically reconfigured, which is an advantage of this example.

4-1-2. Modification 2

[0150] The encoding-decoding dictionary of a quiz question may be configured, for example, by including, in quiz result data contained in play result data transmitted from game terminal **10** to server device **40**, answer data representing an answer received at game terminal **10** for a quiz question having the quiz question ID within the set and by, at server device **40**, associating short encoded data with to-be-encoded data that is the same as answer data representing an answer that has a high frequency of being input and associating long encoded data with to-be-encoded data that is the same as answer data representing an answer that has a low frequency of being input. In short, the encoding-decoding dictionary of a quiz question may be reconfigured so that the length of encoded data will be a length depending on the input frequency of an answer to this quiz question. The encoding-decoding dictionary of a quiz question may thus be configured in a case in which the accuracy rate of the quiz question is less than the standard accuracy rate, or regardless of the accuracy rate of the quiz question.

4-1-3. Modification 3

[0151] For example, a game system may be one in which there is only one quiz question that can be asked. Also, for example, the conversion and reverse-conversion of answer data may be performed without using the encoding-decoding dictionary. In yet another example, it may be configured in such a way that each game terminal **10** stores an encoding-decoding dictionary for all quiz questions. In this case, data to be delivered from server device **40** to each game terminal **10** is not an encoding-decoding dictionary but an accuracy rate of a quiz question. Furthermore, it may be configured in such a manner that each game terminal **10** stores the accuracy rate calculation mandatory data to calculate an accuracy rate of a quiz question at each game terminal **10**.

4-2. Modifications for Association among Plural Quiz Questions

4-2-1. Modification 4

[0152] For example, for a quiz question for which the number of times the quiz question is asked exceeds a certain number, the accuracy rate that is calculated when the number of times the quiz question is a certain number may be maintained. In this case, by setting the certain number as appropriate, a situation is prevented from occurring in which the accuracy rate of the quiz question increases by the same player repeatedly answering the same quiz question.

4-2-2. Modification 5

[0153] For example, in a case in which some or all of the pieces of quiz question data is data representing a character string, the some or all of the pieces of the quiz question may be used as keyword group data. In short, quiz question data may concurrently serve as a quiz-question related keyword group data. Furthermore, the above some or all of the pieces of quiz question data may be limited to data representing a character string that is displayed on screen **121**. Furthermore, the same modifications as these modifications may be made to various types of keyword group data.

4-3. Modifications for Prohibiting the Asking of Inadequate Questions to be Asked

[0154] 4-3-1. Modification 6

[0155] For example, server device **40** may transmit each of plural game terminals **10** data instructing deletion of quiz question data of a quiz question for which the accuracy rate does not fall within an adequate range, so that each of plural game terminals **10** can delete quiz question data in accordance with the data from server device **40**. The terminal-side-do-not-ask list is not necessary in this case.

[0156] Furthermore, for example, quiz question data of a quiz question which game terminal **10** is caused to ask may be transmitted from server device **40** to game terminal **10** for each quiz question. In this case, it may be further modified so that server device **40** does not transmit quiz question data of a quiz question for which the accuracy rate does not fall within the adequate range.

4-3-2. Modification 7

[0157] For example, an adequate range can be determined for each quiz question or may be determined as having the same range for all the questions. Also, for example, the number of quiz questions asked in one quiz game may be one. Furthermore, for example, a quiz data set does not have to be stored in server-side quiz table QT2. For example, the updating of a quiz data set at each game terminal **10** may be performed by a human going to a place at which each game terminal **10** is located.

4-4. Other Modifications

[0158] For example, the content of the quiz game may be dynamically controlled. Specifically, a parent terminal may select, during a multiple-player online game and for every quiz process, a target quiz question from the regulated number of candidate questions, and notifies a result of this selection to every child terminal which is a member, and every member gives a player the target quiz question in a subsequent quiz process in accordance with the notification. If this mode is adopted, it is possible to perform control such that, for example, a quiz question having relatively low difficulty level or having relatively high accuracy rate is determined to be a quiz question for the final quiz in a case in which all players participating in a multiple-player online game have given incorrect answers for all the questions so far and are coming up to the final quiz, if the regulated number is determined to be sufficiently large. According to this control, there will be a lower probability of impairing game-worthiness. Furthermore, the probability is reduced of the occurrence of situations in which the player satisfaction is extremely low (no correct answer).

[0159] For example, the game system may have a configuration in which freely-selected game terminals do not have a parent-child (master-client) relationship, but communicate with one another on an equal basis. In this case, the communication address of each member is shared by all the members. Furthermore, for example, a home-use game console may be used as game terminal **10**. In this case, game terminal **10** is placed in households. Also, for example, a portable type game device may be used as game terminal **10**. In this case, game terminal **10** is carried by a player.

[0160] In the above embodiment, description was given of a case in which computer program P1 is executed by processor **150** of game terminal **10**, but by executing the computer program by a general-use computer, the processor of the computer may be caused to execute the process of each of the above-described embodiments.

[0161] Additionally, in the above embodiment, description was given of a case in which computer program P**1** is stored in non-volatile area RT**2** of terminal storage device **160** in server device **40**, but the computer program may be recorded, as data loadable onto a memory of a computer, on a data carrier such as a floppy (registered trademark) disk, a CD-ROM, or any other computer-readable recording medium and be distributed. The data carrier may be a data connection, such as for transmitting signals representing computer program P**1** via a telephone line or a wireless connection. For example, computer program P**1** may be downloaded through communication interface **180** by connecting to the Internet **2**.

1. A game terminal for enabling a player to play a quiz game including a quiz, the game terminal comprising:

a quiz processor that, in the quiz, asks the player a quiz question, that receives an answer for the quiz question, and that informs a result of the quiz after the quiz is completed;

a converter that converts answer data representing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct;

a transmitter that transmits to another game terminal the answer data converted by the converter;

a receiver that receives the answer data from another game terminal; and

a reverse-converter that reverse-converts the answer data received by the receiver.

2. A game terminal according to claim **1**,

the converter comprising:

an encoding-decoding dictionary storage device that stores an encoding-decoding dictionary in which a to-be-encoded bit string and an encoded bit string are stored in a correlated manner for each of plural pieces of data that are different from one another; and

an encoder that encodes the answer data by using the encoding-decoding dictionary stored in the encoding-decoding dictionary storage device,

wherein the answer data encoded by the encoder is treated as the answer data converted by the converter.

3. A game terminal according to claim **1**,

the converter comprising:

an encoding-decoding dictionary storage device that stores an encoding-decoding dictionary in which a to-be-encoded bit string and an encoded bit string are stored in a correlated manner for each of plural pieces of data that are different from one another;

an encoder that encodes the answer data by using the encoding-decoding dictionary stored in the encoding-decoding dictionary storage device; and

an extractor that extracts a portion that is included in the bit string of the answer data encoded by the encoder and that is not included in another encoded bit string;

wherein the portion extracted by the extractor is treated as the answer data converted by the converter.

4. A game system including plural game terminals, each for enabling a player to play a quiz game including a quiz,

each of the plural game terminals comprising:

a quiz processor that, in the quiz, asks the player a quiz question, that receives an answer for the quiz question, and that informs a result of the quiz after the quiz is completed;

a converter that converts answer data showing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct;

a transmitter that transmits to another game terminal the answer data converted by the converter;

a receiver that receives the answer data from another game terminal; and

a reverse-converter that reverse-converts the answer data received by the receiver.

5. A game system according to claim **4**,

further comprising a server device for communicating with each of the plural game terminals,

the server device comprising:

a calculation mandatory data storage device that stores, for each quiz question, data of parameters mandatory for accuracy rate calculation as calculation mandatory data; and

a calculation mandatory data updater that updates, based on the result of the quiz, the calculation mandatory data stored in the calculation mandatory data storage device,

wherein the converter converts answer data showing the answer in such a manner that a bit string of the answer data after the conversion will be the shortest when the answer is correct, the conversion being performed in a case in which an accuracy rate of a quiz question calculated based on the calculation mandatory data stored in the calculation mandatory data storage device is equal to or greater than a standard accuracy rate.

6. A computer program for running on a computer that enables a player to play a quiz game including a quiz, the computer program, when run on the computer, at least including computer-readable instructions for performing the steps of:

processing a game, the processing including asking the player a quiz question, receiving an answer for the quiz question, and informing a result of the quiz after the quiz is completed;

converting answer data showing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct;

transmitting to another game terminal the answer data converted in the converting step;

receiving the answer data from another game terminal; and

reverse-converting the answer data received in the receiving step.

7. A computer-readable recording medium that stores a computer program for running on a computer that enables a player to play a quiz game including a quiz, the program, when run on the computer, at least including computer readable instructions for performing the steps of:

processing a game, the processing including asking the player a quiz question, receiving an answer for the quiz question, and informing a result of the quiz after the quiz is completed;

converting answer data showing the answer in such a manner that a bit string of the answer data will be the shortest after the conversion when the answer is correct;

transmitting to another game terminal the answer data converted in the converting step;

receiving the answer data from another game terminal; and

reverse-converting the answer data received in the receiving step.

* * * * *