(12) UK Patent Application (19) GB (11) 2 163 929 A

(43) Application published 5 Mar 1986

(21) Application No 8520848

(22) Date of filing 20 Aug 1985

(30) Priority data

(31) 59/182080 (32) 30 Aug 1984 (33) JP

(51) INT CL[4]
A63F 9/22

(52) Domestic classification
H4T 4R BRA
U1S 1172 H4T

(56) Documents cited
GB 1567967

(58) Field of search
H4T

(71) Applicant
Casio Computer Co. Ltd. (Japan),
6-1 2-chome Nishi-shinjuku, Shinjuku-ku, Tokyo, Japan

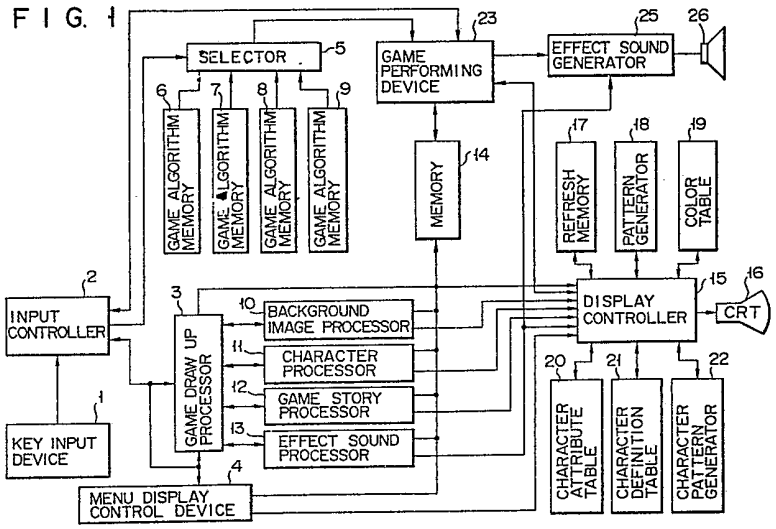
(72) Inventors
Satoshi Aso,
Hiroshi Kurosawa,
Minoru Sasakura,
Ryuji Saito

(74) Agent and/or Address for Service
A. A. Thornton & Co.,
Northumberland House, 303-306 High Holborn,
London WC1V 7LE

(54) Electronic game apparatus

(57) An electronic game apparatus using an animation display 16, comprises a game for setting unit 1, 2, 5-9 for selectively designating basic game algorithm data of a desired game of a predetermined form which has been prepared in advance, a background image setting unit 1-3, 10, 14, 15, 17-19 for selectively setting background image data to be added to the designated basic game algorithm data, a character setting unit 1-3, 11, 14, 15, 20-22 for selectively setting character data to be added to the designated basic game algorithm data, a game story setting unit (1-3, 12, 14, 15) for selectively setting the game story data to be added to the designated basic game algorithm data, and a game performing unit 1, 2, 23 for combining the background image data, the character data and the game story data to the designated basic game algorithm data so as to be displayed on the animation display 16 thus performing the game in accordance with the predetermined form.

FIG. 1



GB 2 163 929 A

2163929

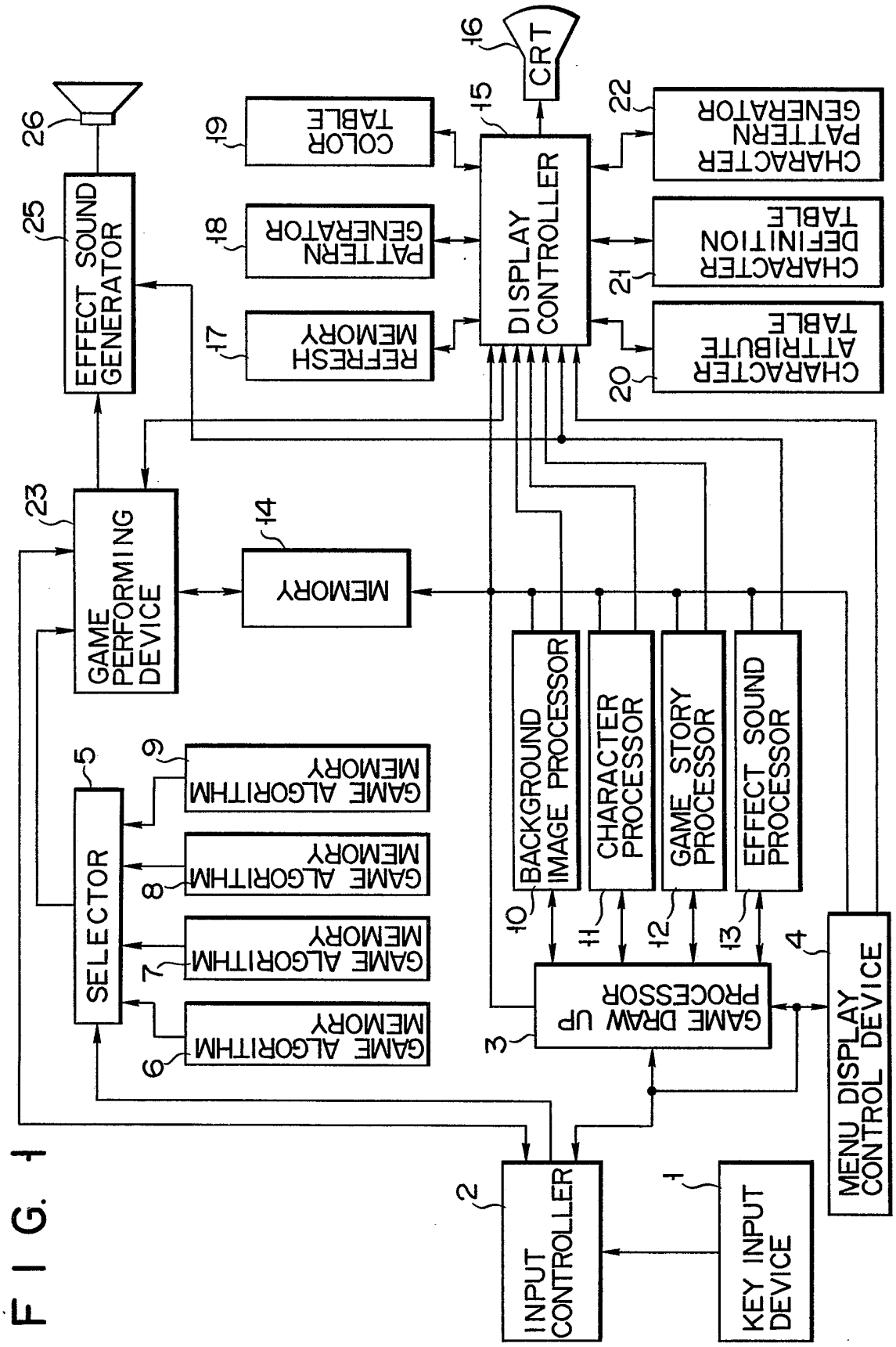


FIG. 1

2163929

FIG. 2A

0	1	1	0	1	1	0	0

FIG. 2B

COLOR OF "1"	COLOR OF "0"

FIG. 3

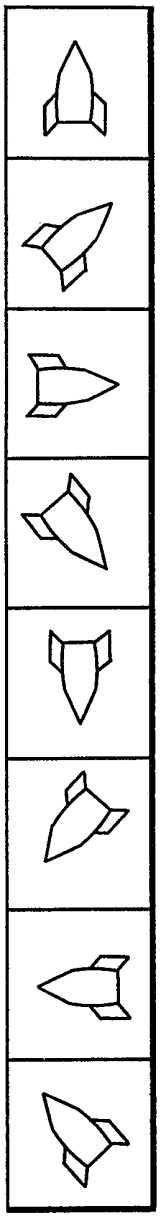
20

20a	ATTRIBUTE
20b	CHARACTER NAME
20c	UPWARD SURFACE (TABLE #)
20d	UPWARD BACK
20e	UNDER SURFACE
20f	UNDER BACK
20g	LEFT SURFACE
20h	LEFT BACK
20i	RIGHT SURFACE
20j	RIGHT BACK
20k	INFORMATION ADDRESS

FIG. 4

000	SELF
001	ARMS OF SELF
010	RIVAL
011	ARMS OF RIVAL
100	DEFEAT FORM

FIG. 5



2163929

FIG. 6

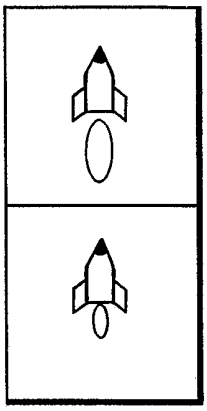


FIG. 8

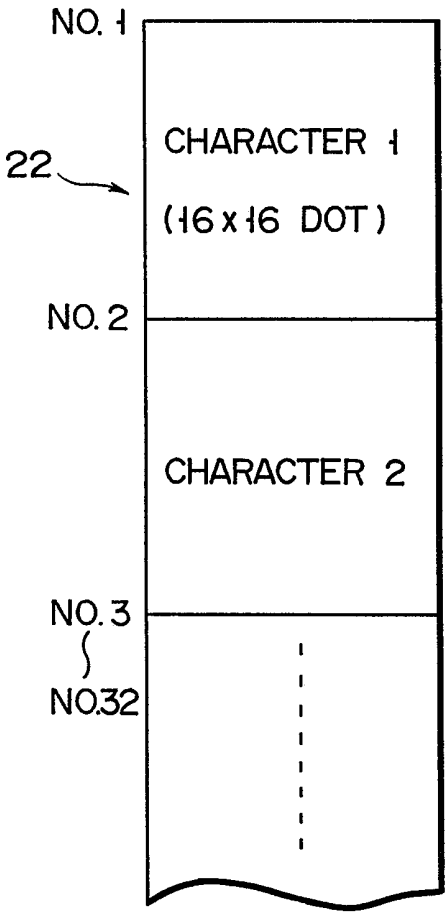
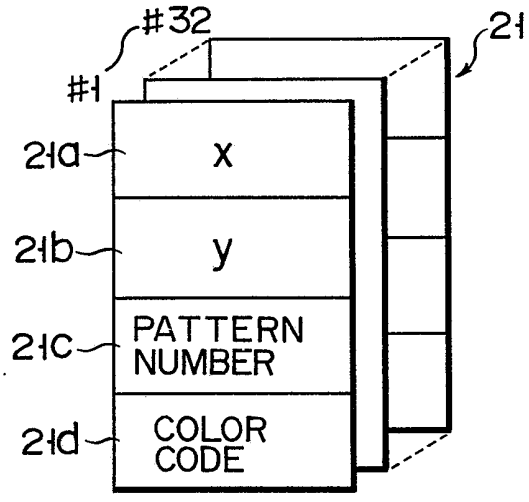


FIG. 7



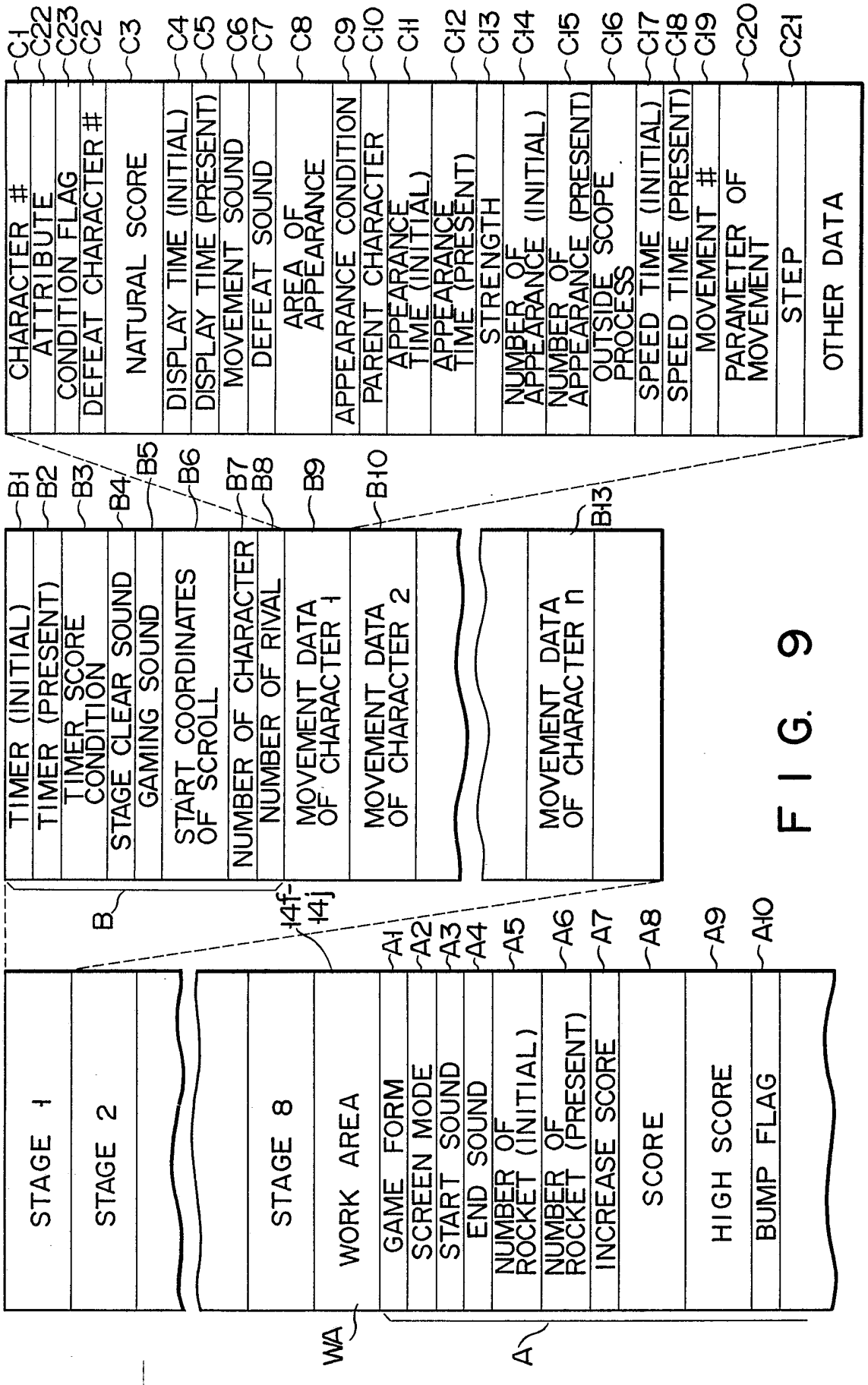

STAGE 1
STAGE 2
STAGE 8
WORK AREA
GAME FORM
SCREEN MODE
START SOUND
END SOUND
NUMBER OF ROCKET (INITIAL)
NUMBER OF ROCKET (PRESENT)
INCREASE SCORE
SCORE
HIGH SCORE
BUMP FLAG
TIMER (INITIAL)
TIMER (PRESENT)
TIMER SCORE CONDITION
STAGE CLEAR SOUND
GAMING SOUND
START COORDINATES OF SCROLL
NUMBER OF CHARACTER
NUMBER OF RIVAL
MOVEMENT DATA OF CHARACTER 1
MOVEMENT DATA OF CHARACTER 2
MOVEMENT DATA OF CHARACTER n
CHARACTER #
ATTRIBUTE
CONDITION FLAG
DEFEAT CHARACTER #
NATURAL SCORE
DISPLAY TIME (INITIAL)
DISPLAY TIME (PRESENT)
MOVEMENT SOUND
DEFEAT SOUND
AREA OF APPEARANCE
APPEARANCE CONDITION
PARENT CHARACTER
APPEARANCE TIME (INITIAL)
APPEARANCE TIME (PRESENT)
STRENGTH
NUMBER OF APPEARANCE (INITIAL)
NUMBER OF APPEARANCE (PRESENT)
OUTSIDE SCOPE PROCESS
SPEED TIME (INITIAL)
SPEED TIME (PRESENT)
MOVEMENT #
PARAMETER OF MOVEMENT
STEP
OTHER DATA
WA
A
B
14f-14j
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B13
C1
C22
C23
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21

FIG. 9

2163929

FIG. 10

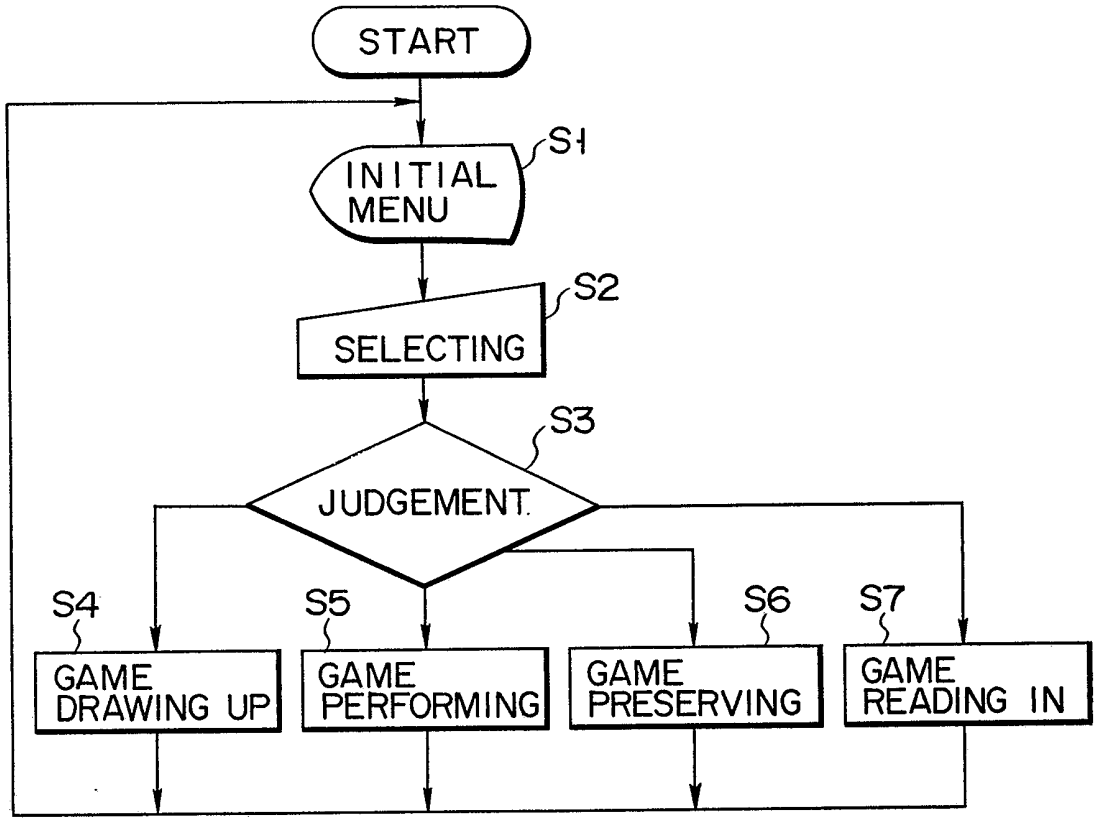


FIG. 11

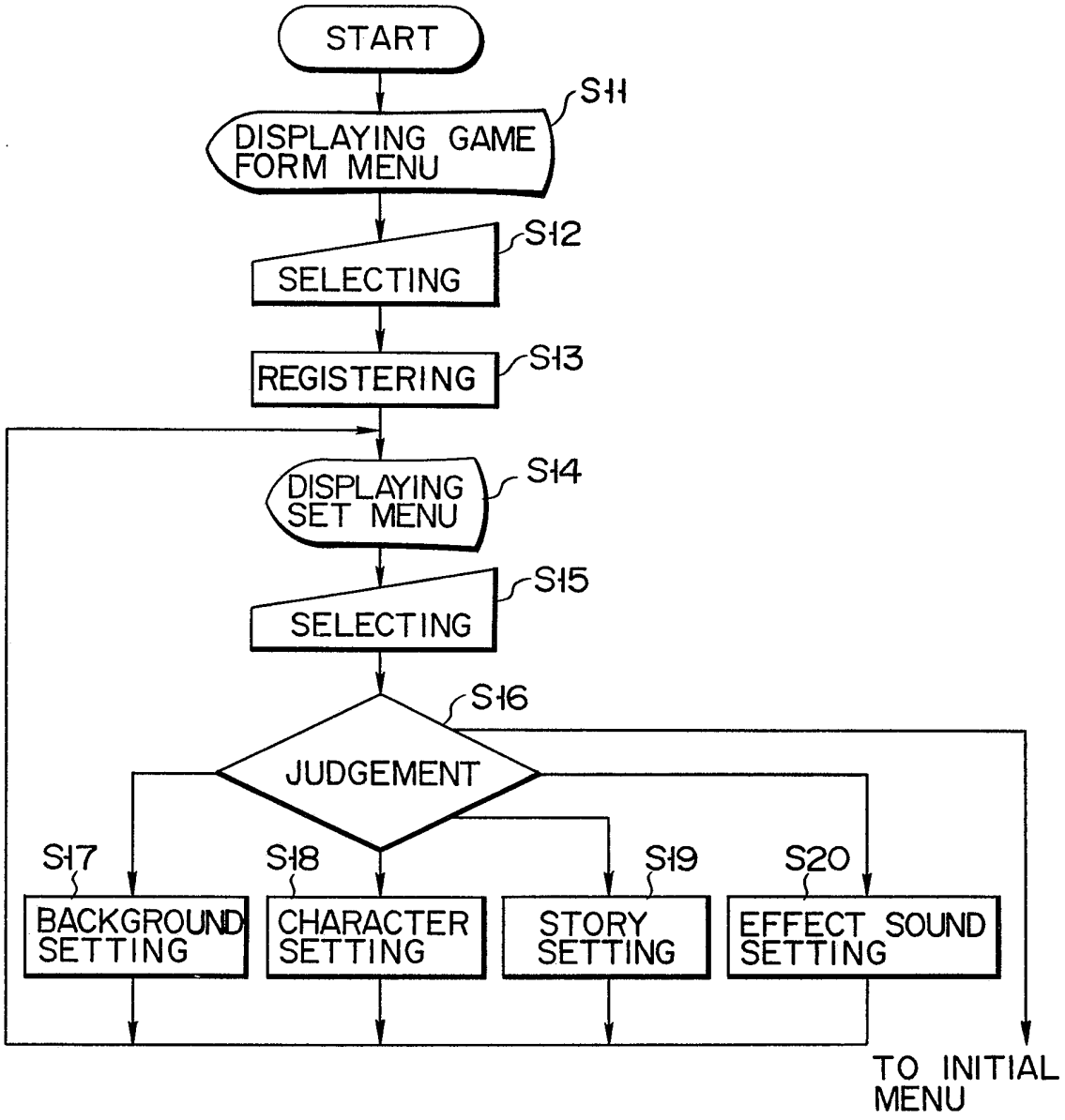
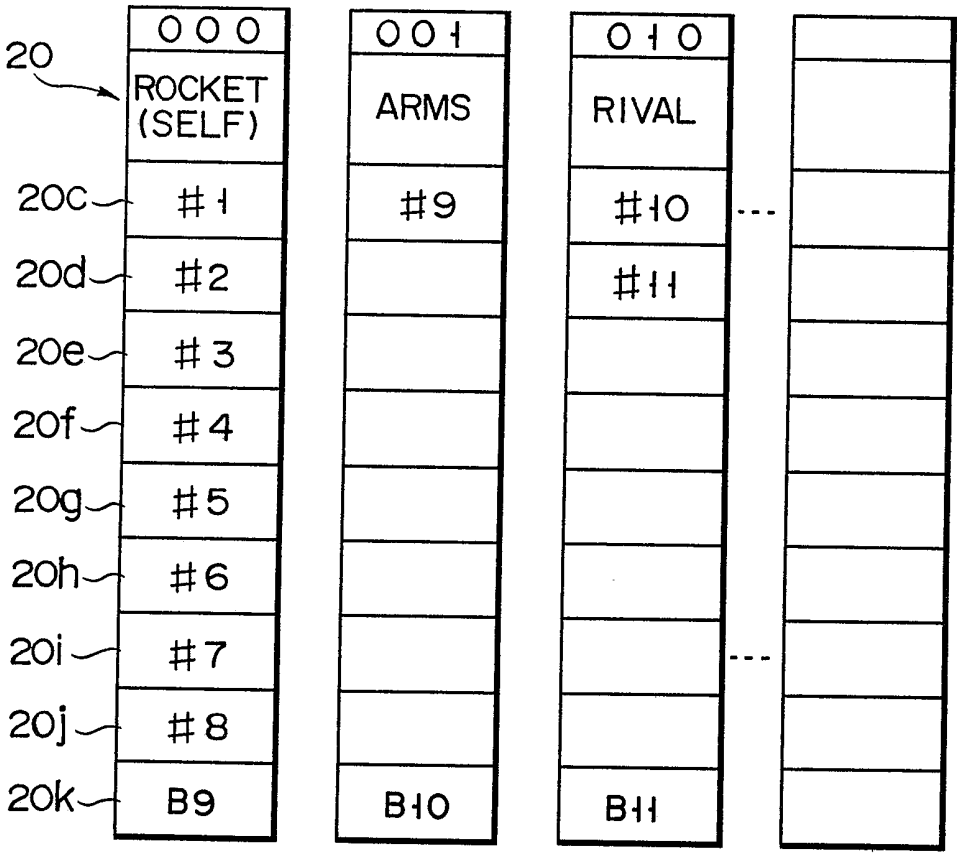

START
S11
DISPLAYING GAME FORM MENU
S12
SELECTING
S13
REGISTERING
S14
DISPLAYING SET MENU
S15
SELECTING
S16
JUDGEMENT
S17
BACKGROUND SETTING
S18
CHARACTER SETTING
S19
STORY SETTING
S20
EFFECT SOUND SETTING
TO INITIAL MENU

FIG. 12


20
000
ROCKET (SELF)
#1
#2
#3
#4
#5
#6
#7
#8
B9
20c
20d
20e
20f
20g
20h
20i
20j
20k
001
ARMS
#9
B10
010
RIVAL
#10
#11
B11

2163929

FIG. 13A

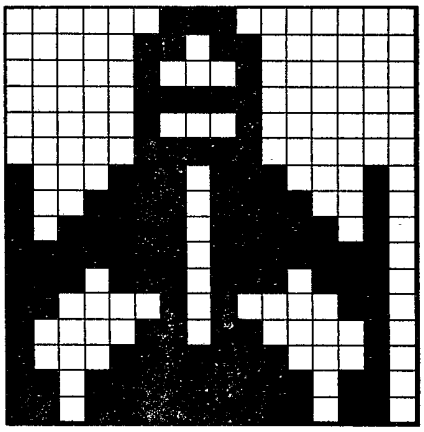


FIG. 13B

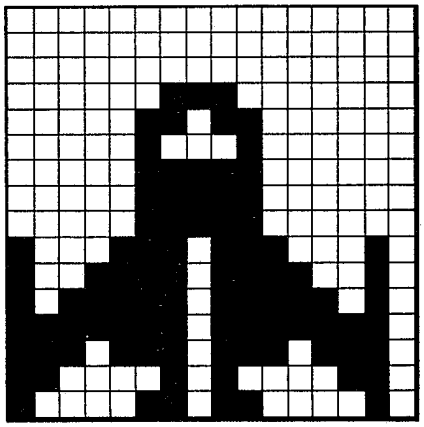


FIG. 13C

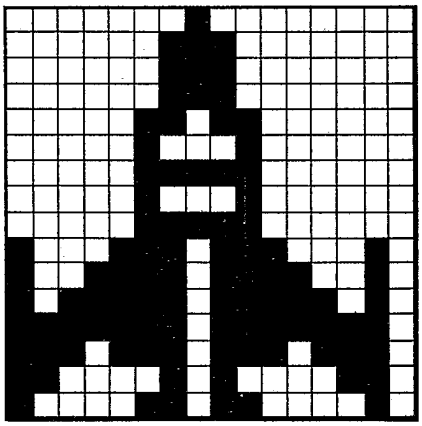
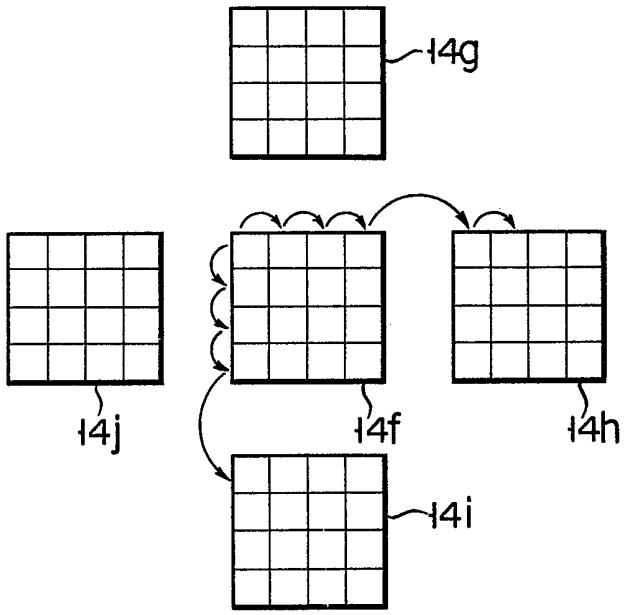


FIG. 14



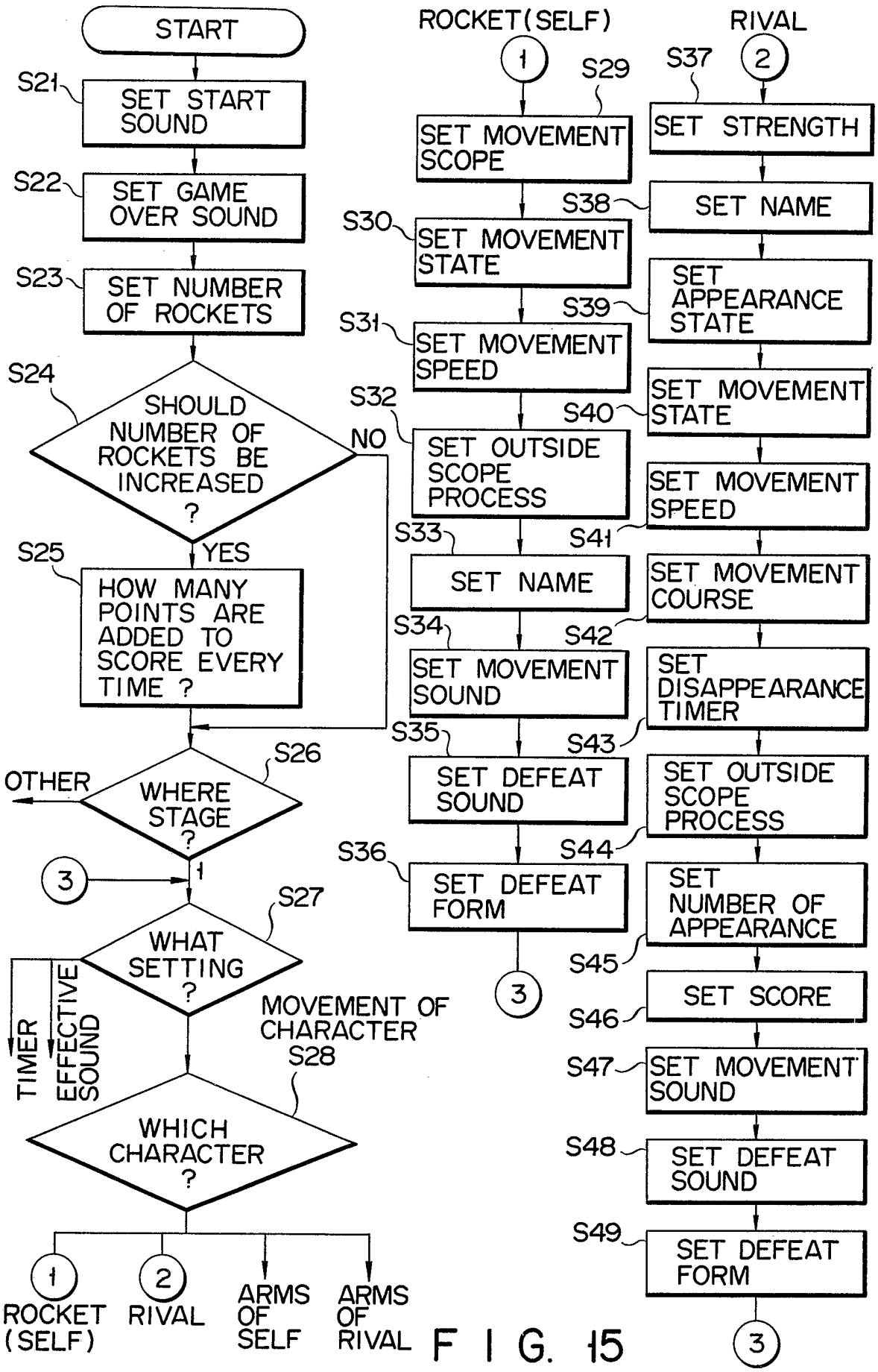
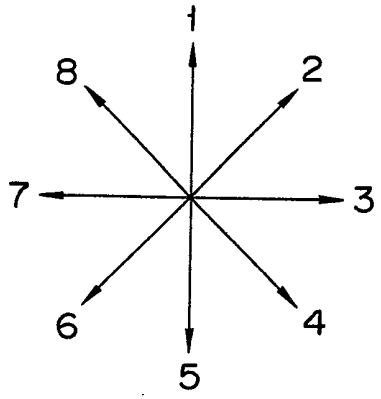

START
S21
SET START SOUND
S22
SET GAME OVER SOUND
S23
SET NUMBER OF ROCKETS
S24
SHOULD NUMBER OF ROCKETS BE INCREASED ?
NO
YES
S25
HOW MANY POINTS ARE ADDED TO SCORE EVERY TIME ?
S26
OTHER
WHERE STAGE ?
3
1
S27
WHAT SETTING ?
TIMER
EFFECTIVE SOUND
MOVEMENT OF CHARACTER
S28
WHICH CHARACTER ?
1
ROCKET (SELF)
2
RIVAL
ARMS OF SELF
ARMS OF RIVAL
ROCKET(SELF)
1
S29
SET MOVEMENT SCOPE
S30
SET MOVEMENT STATE
S31
SET MOVEMENT SPEED
S32
SET OUTSIDE SCOPE PROCESS
S33
SET NAME
S34
SET MOVEMENT SOUND
S35
SET DEFEAT SOUND
S36
SET DEFEAT FORM
3
RIVAL
S37
2
SET STRENGTH
S38
SET NAME
S39
SET APPEARANCE STATE
S40
SET MOVEMENT STATE
S41
SET MOVEMENT SPEED
S42
SET MOVEMENT COURSE
S43
SET DISAPPEARANCE TIMER
S44
SET OUTSIDE SCOPE PROCESS
S45
SET NUMBER OF APPEARANCE
S46
SET SCORE
S47
SET MOVEMENT SOUND
S48
SET DEFEAT SOUND
S49
SET DEFEAT FORM
3

FIG. 15

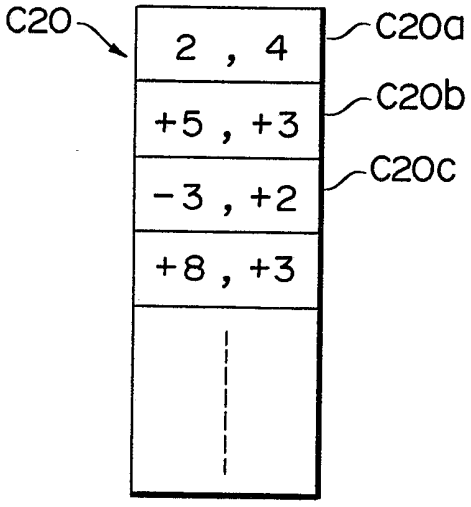
F I G. 16A



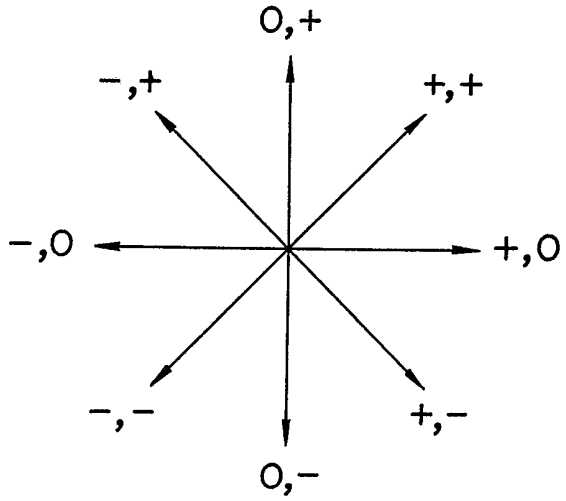
F I G. 16B

	1	2	3	4	5	6	7	8
ROCKET	1	1	1	1	1	1	1	1
RIVAL	1	0	0	0	1	0	0	0

F I G. 18

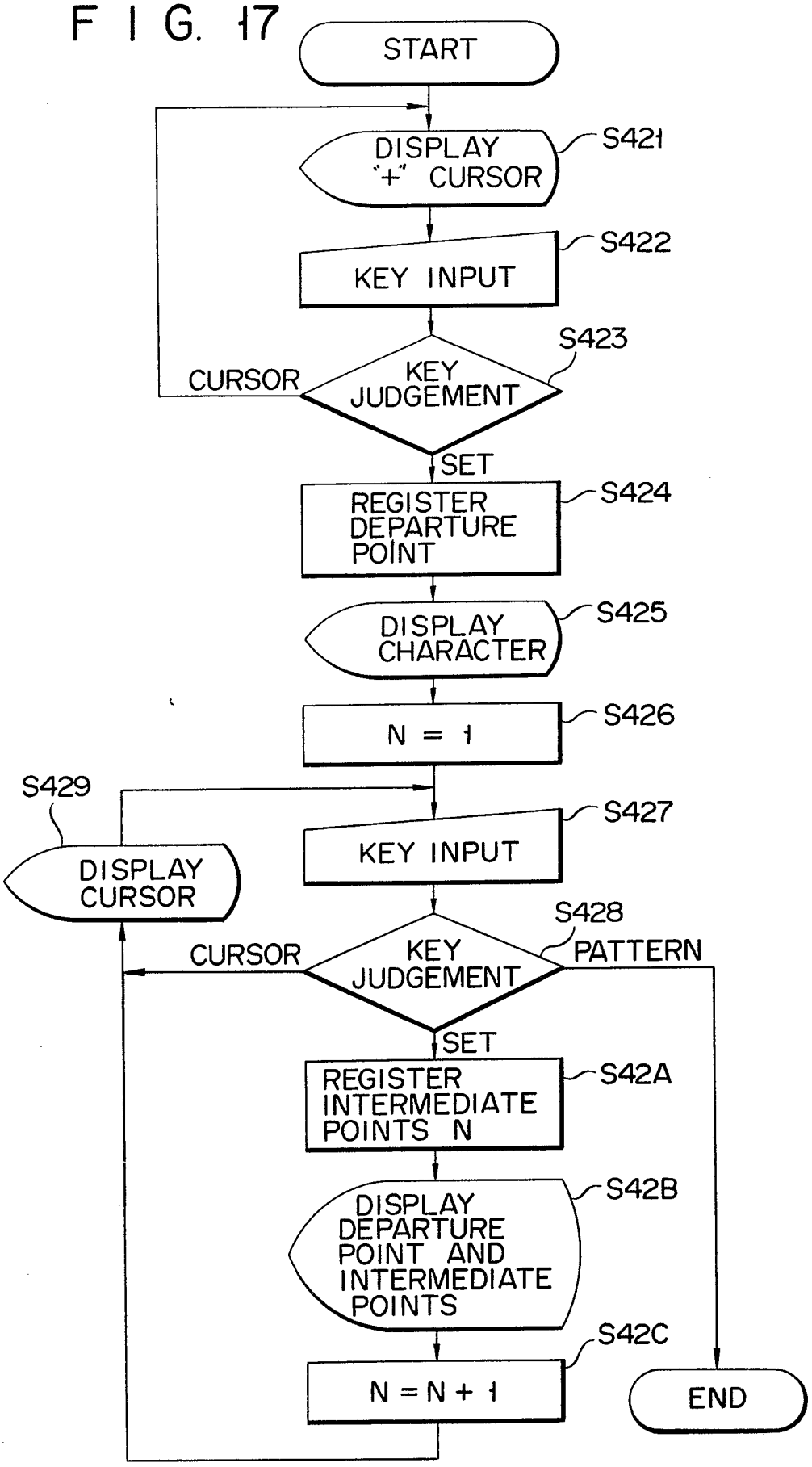


F I G. 23



2163929

FIG. 17



2163929

FIG. 19

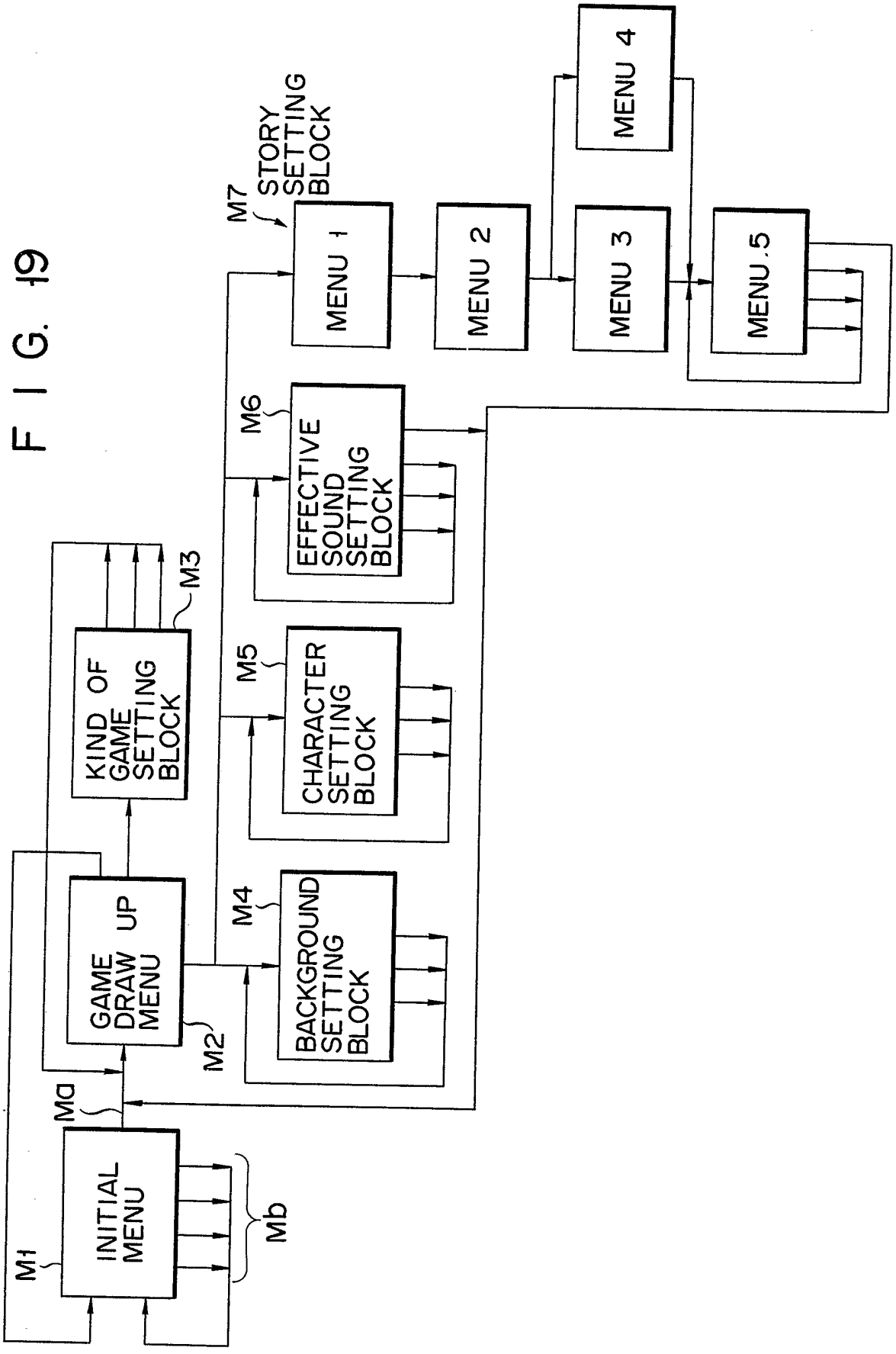


FIG. 20

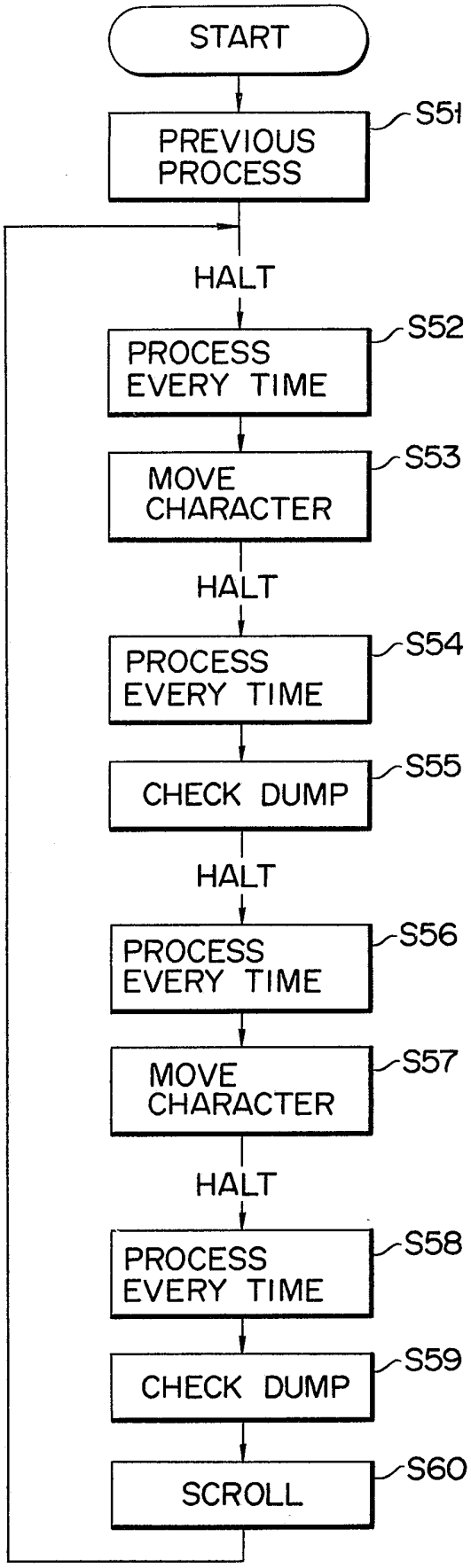
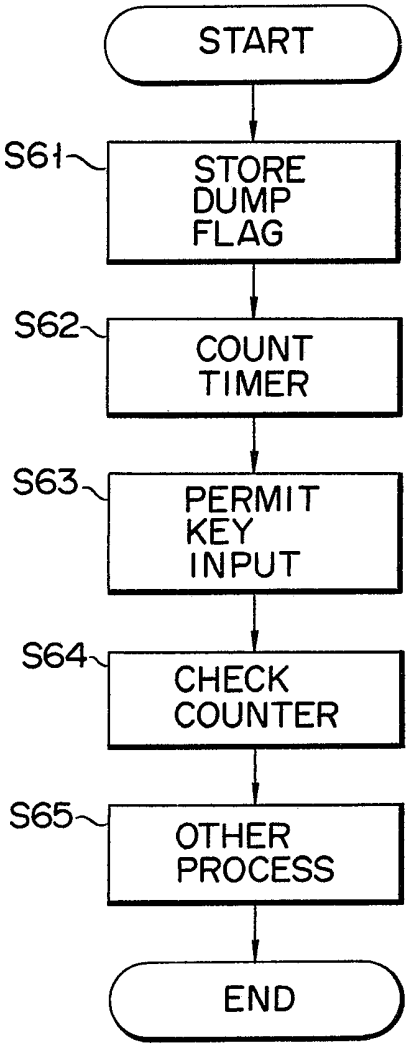
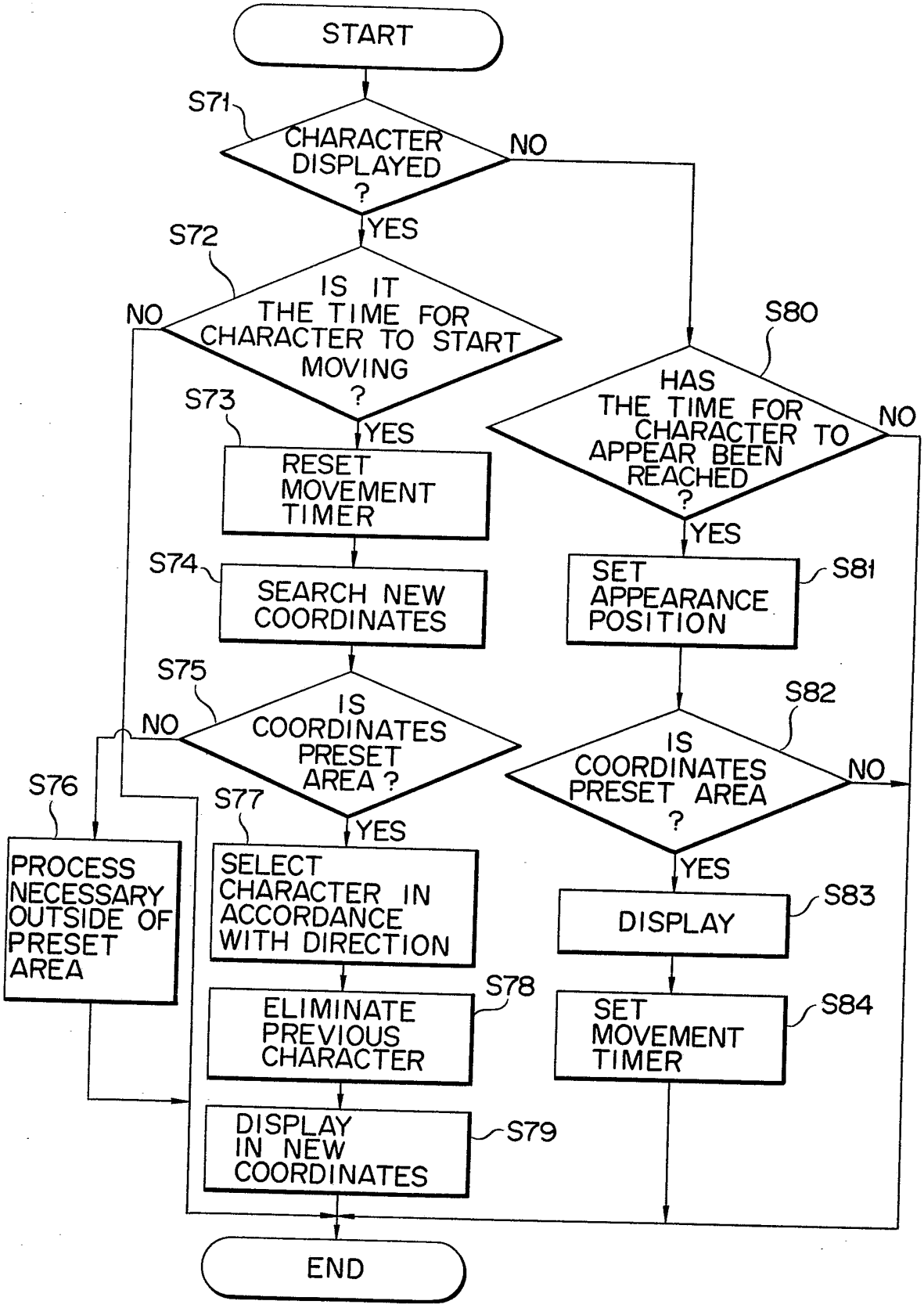

START
PREVIOUS PROCESS
S51
HALT
PROCESS EVERY TIME
S52
MOVE CHARACTER
S53
HALT
PROCESS EVERY TIME
S54
CHECK DUMP
S55
HALT
PROCESS EVERY TIME
S56
MOVE CHARACTER
S57
HALT
PROCESS EVERY TIME
S58
CHECK DUMP
S59
SCROLL
S60

FIG. 21


START
S61
STORE DUMP FLAG
S62
COUNT TIMER
S63
PERMIT KEY INPUT
S64
CHECK COUNTER
S65
OTHER PROCESS
END

2163929

FIG. 22



2163929

FIG. 24

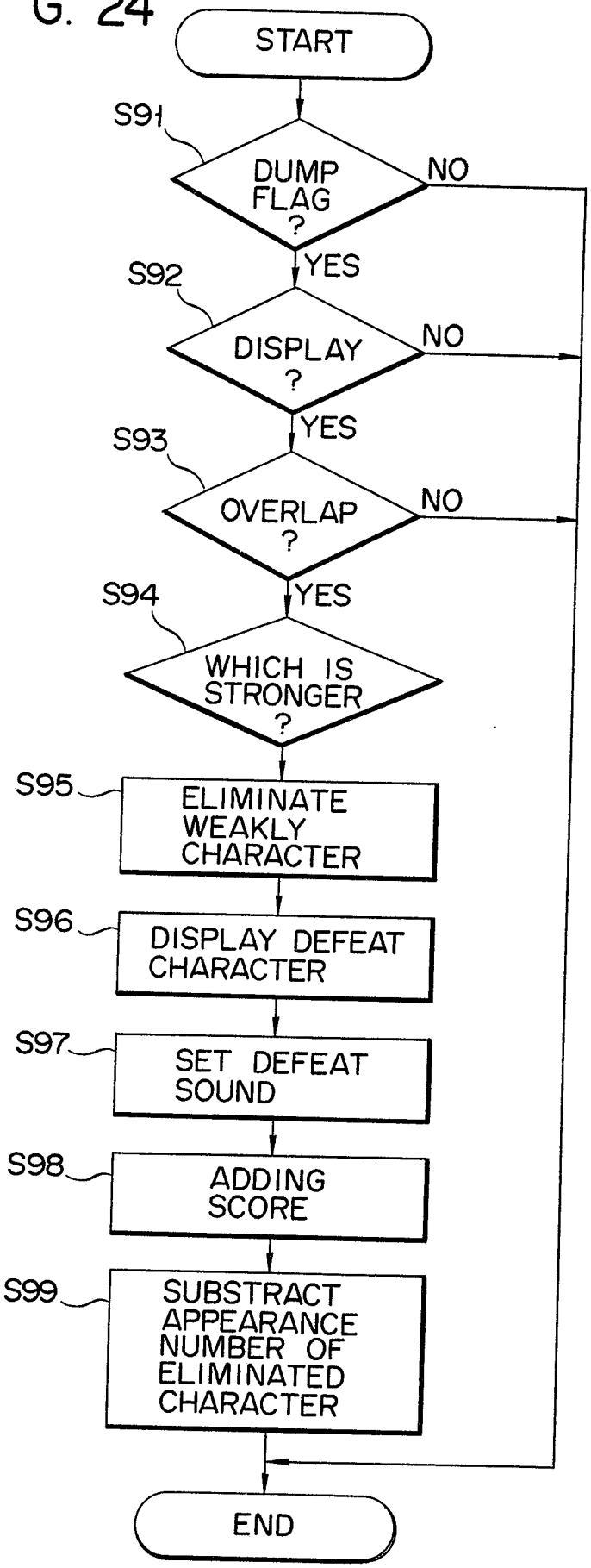


FIG. 25A

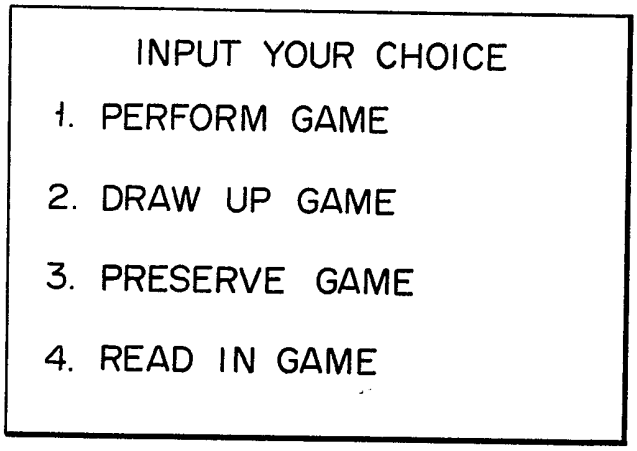


FIG. 25B

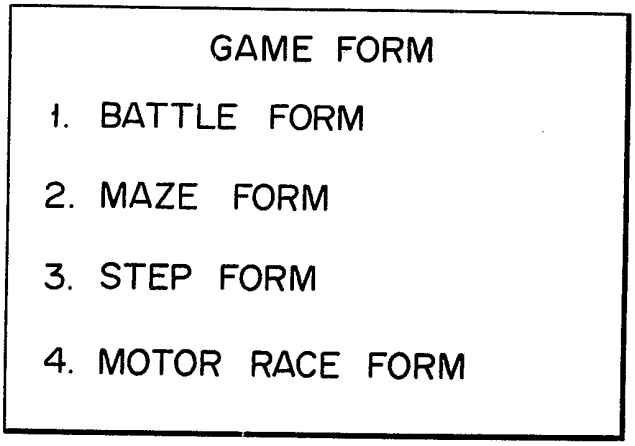


FIG. 25C

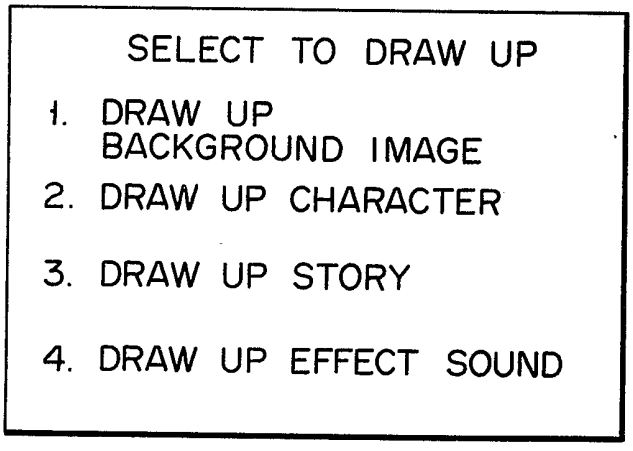


FIG. 25D

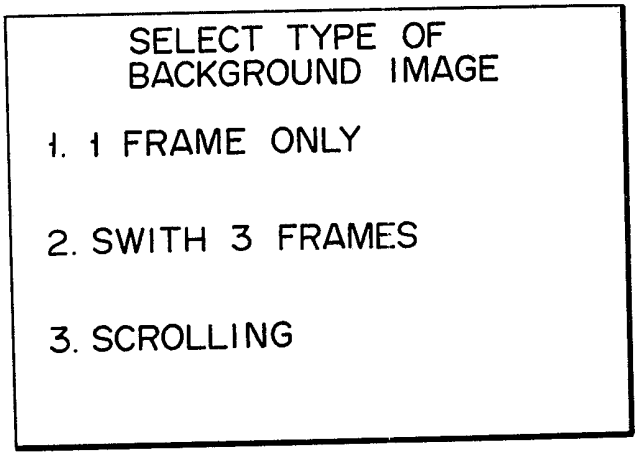


FIG. 25E

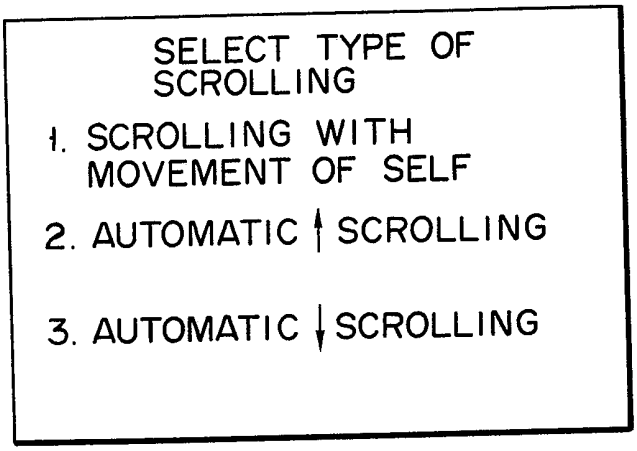


FIG. 26A

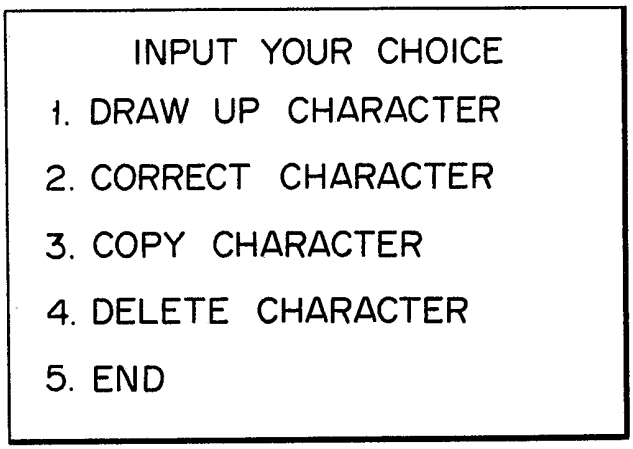


FIG. 26B

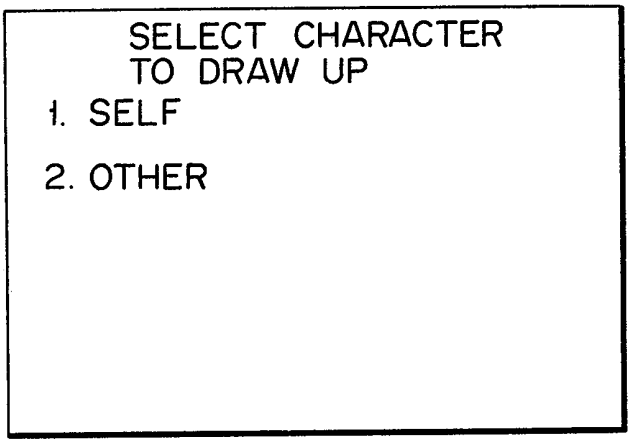


FIG. 26C

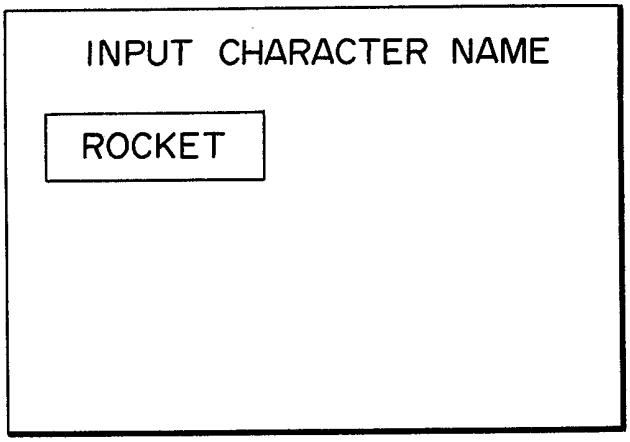


FIG. 26D

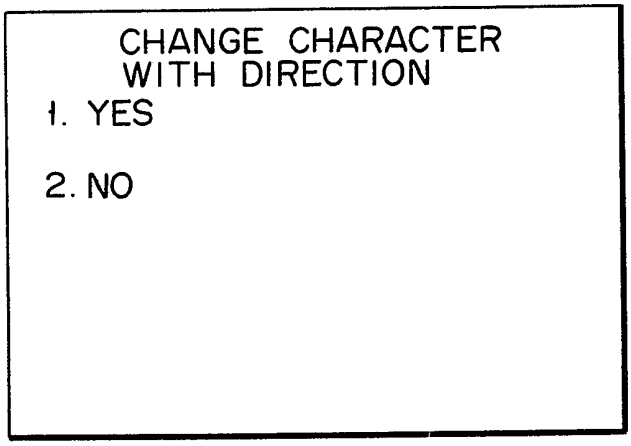


FIG. 27A

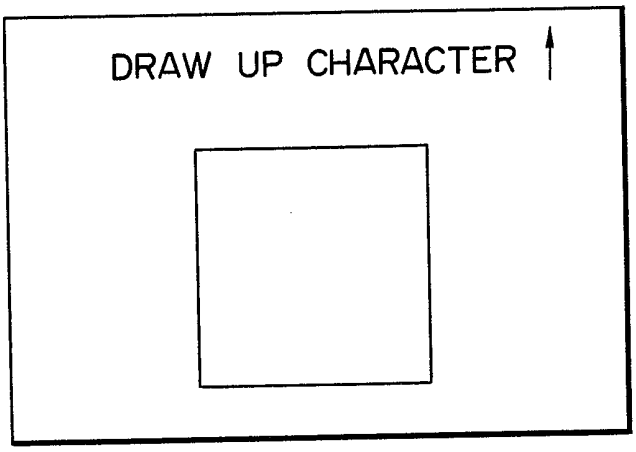


FIG. 27B

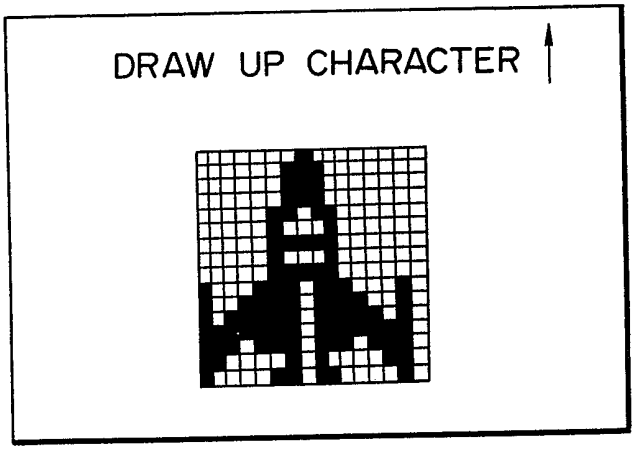


FIG. 27C

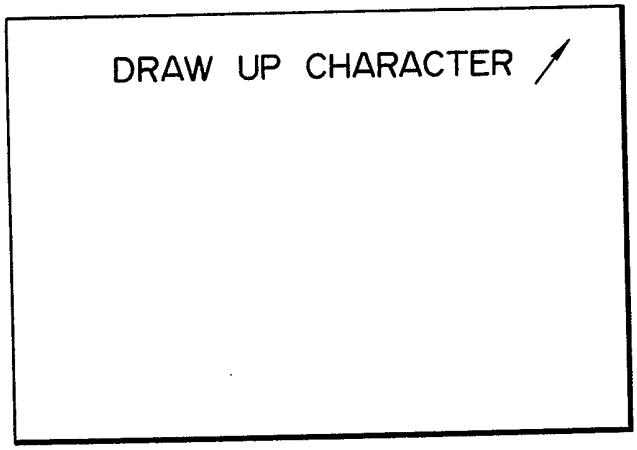


FIG. 28A

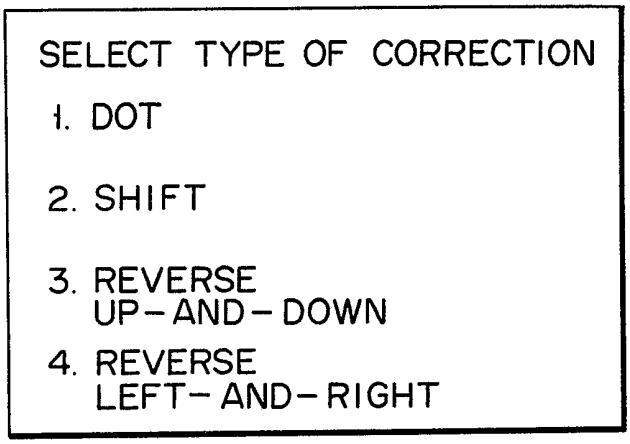


FIG. 28B

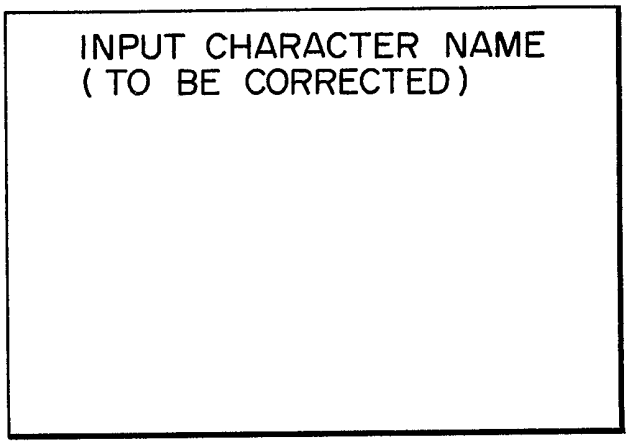


FIG. 29A

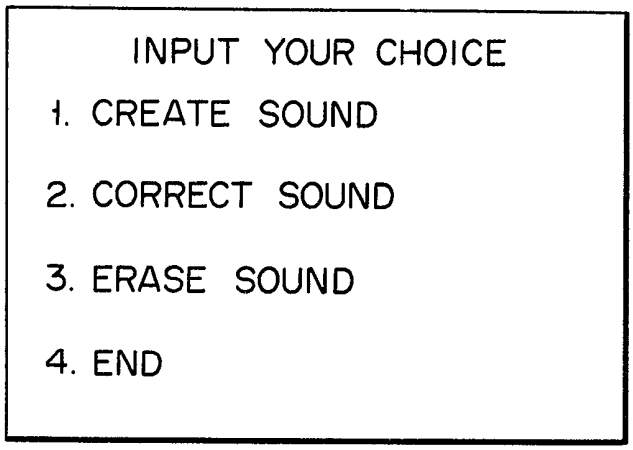


FIG. 29B

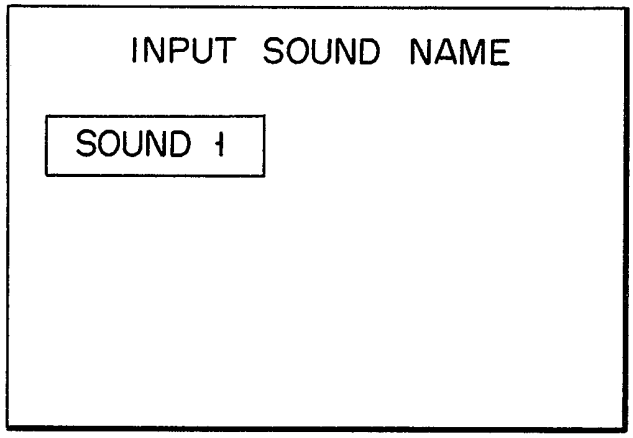


FIG. 29C

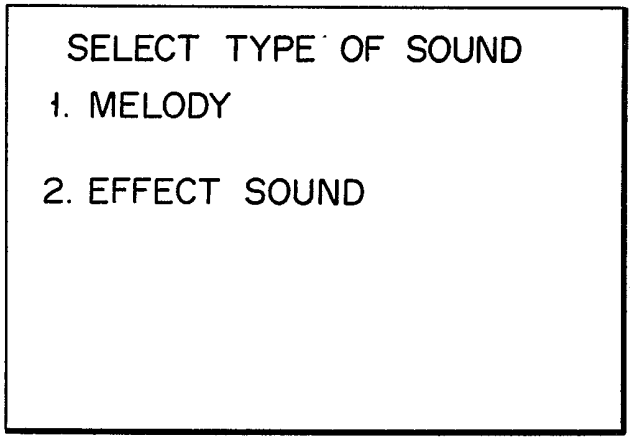


FIG. 29D

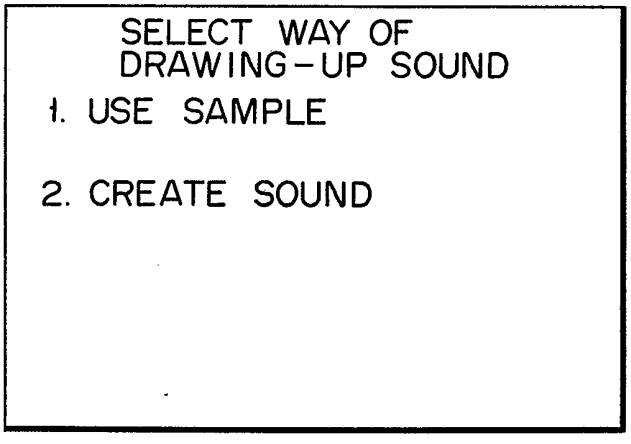


FIG. 29E

INPUT YOUR CHOICE

1. MELODY OF "MARRIAGE"
2. MELODY OF "BATTLE"
3. MELODY OF "CHARGE"
4. MELODY OF "TITLE"

FIG. 29F

INPUT YOUR CHOICE

1. MUSIC PIECE
2. TEMPO
3. REPEAT
4. CHORD
5. END

TEMPO = 5
REPEAT = OFF
CHORD = OFF

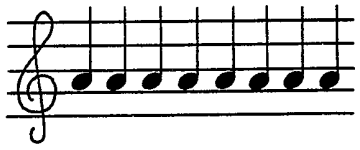


FIG. 29G

SELECT SOUND NAME

1. SOUND 1
2. SOUND 2
3. SOUND 3

FIG. 29H

INPUT YOUR CHOICE

1. MUSIC PIECE
2. PLAY BACK SOUND
3. TEMPO
4. REPEAT
5. CHORD
6. END

TEMPO = 7
REPEAT = OFF
CHORD = ON

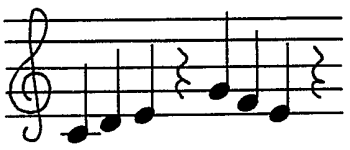


FIG. 29I

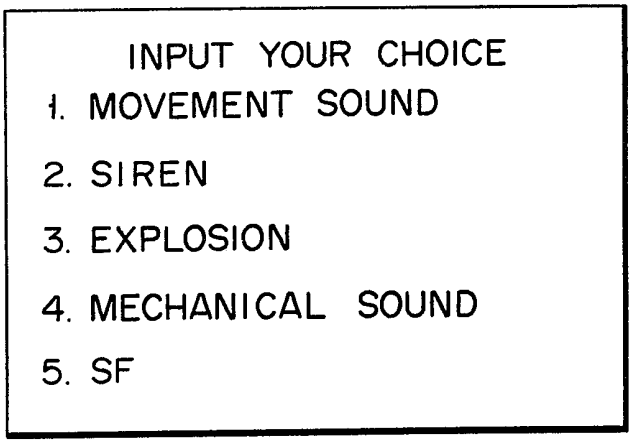
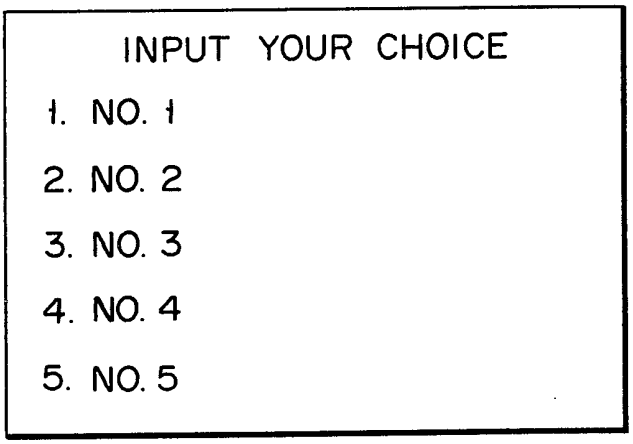


FIG. 29J



2163929

FIG. 30A

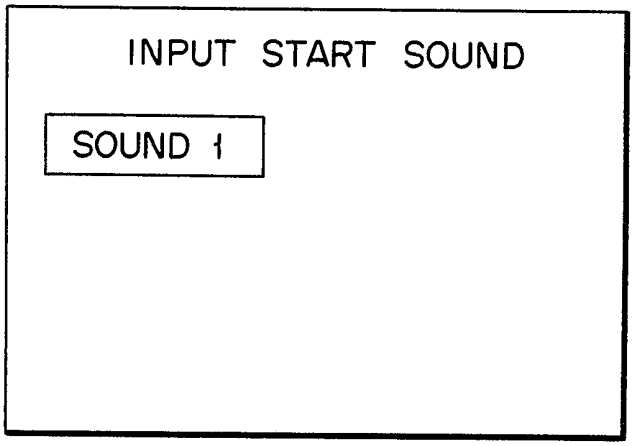


FIG. 30B

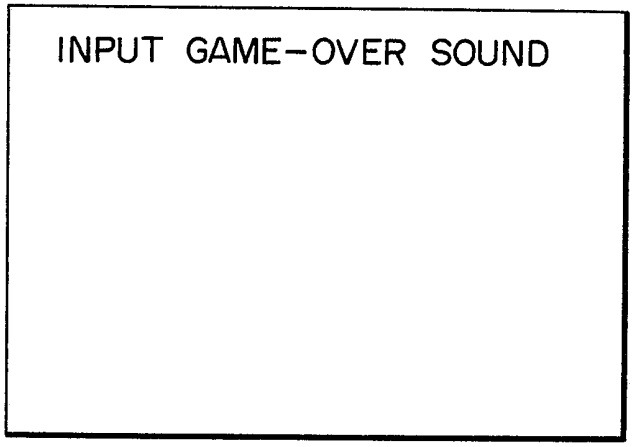


FIG. 30C

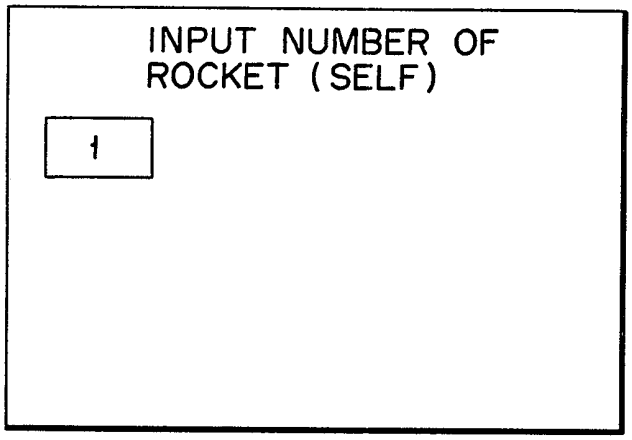
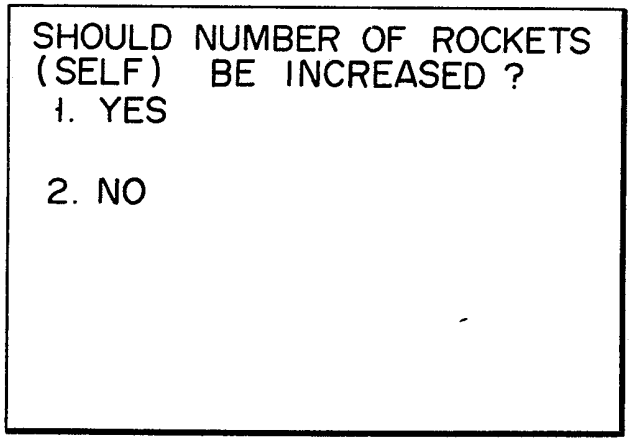


FIG. 30D



2163929

FIG. 31A

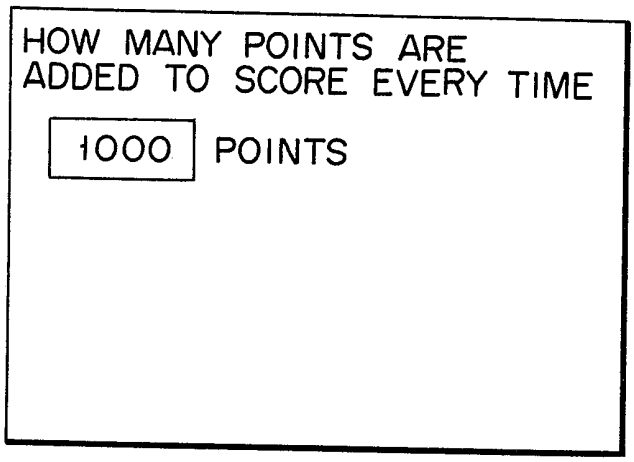


FIG. 31B

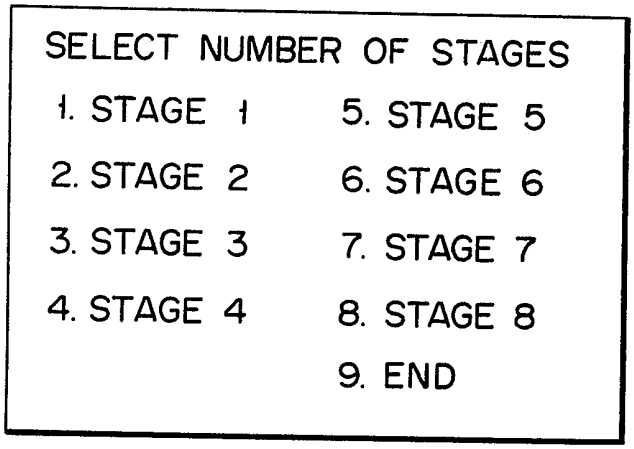


FIG. 31C

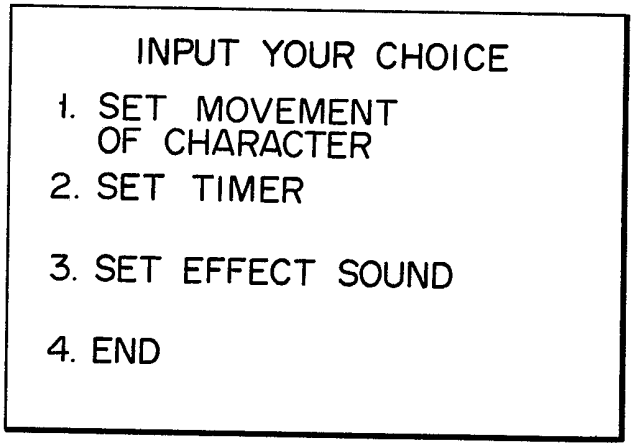


FIG. 31D

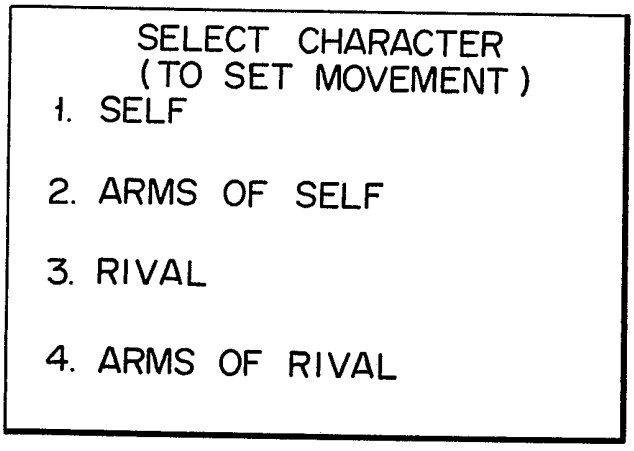


FIG. 32A

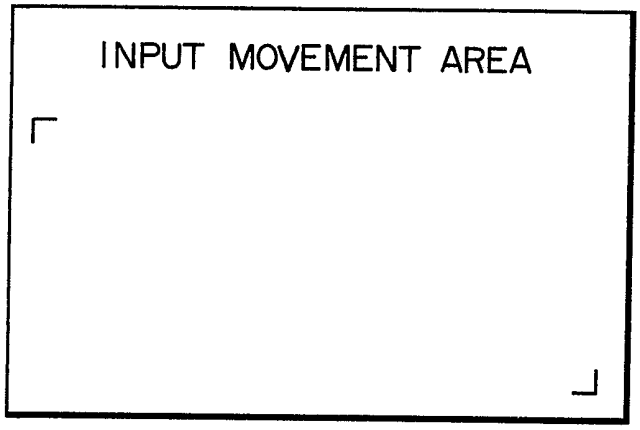


FIG. 32B

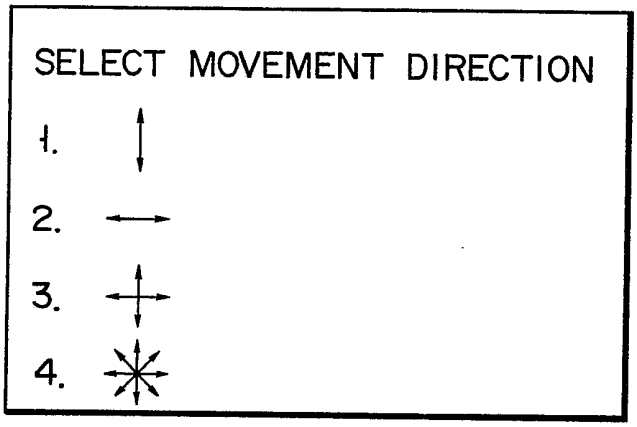


FIG. 32C

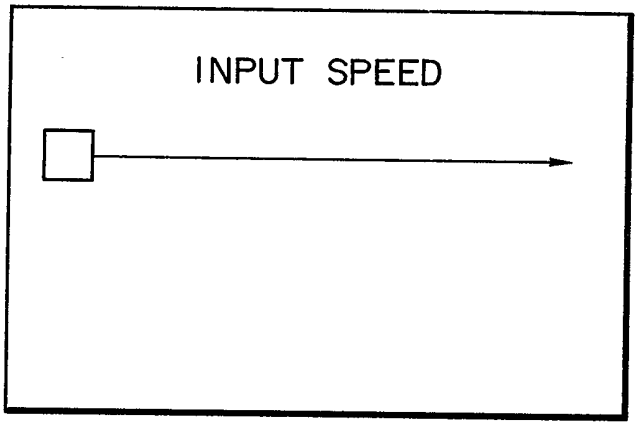
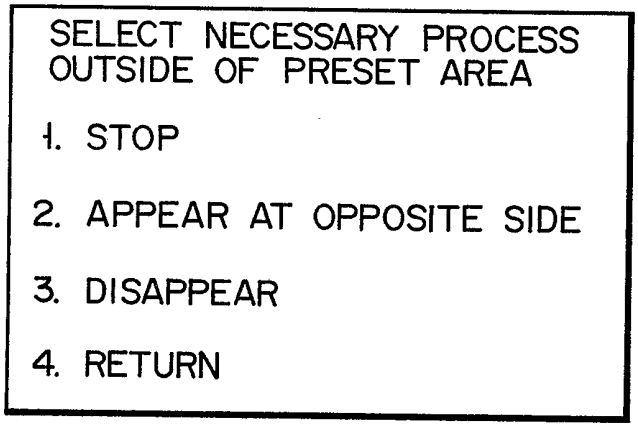


FIG. 32D



2163929

FIG. 33A

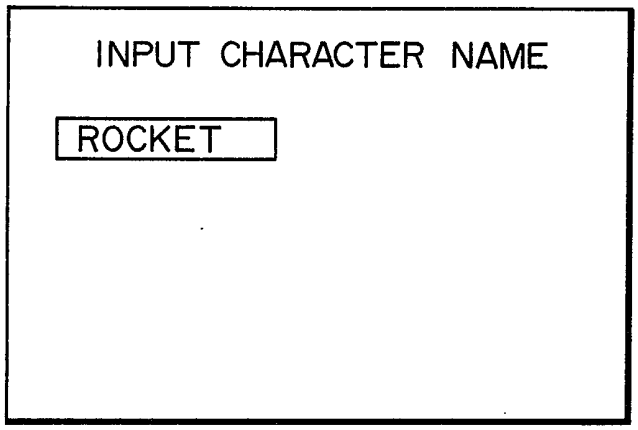


FIG. 33B

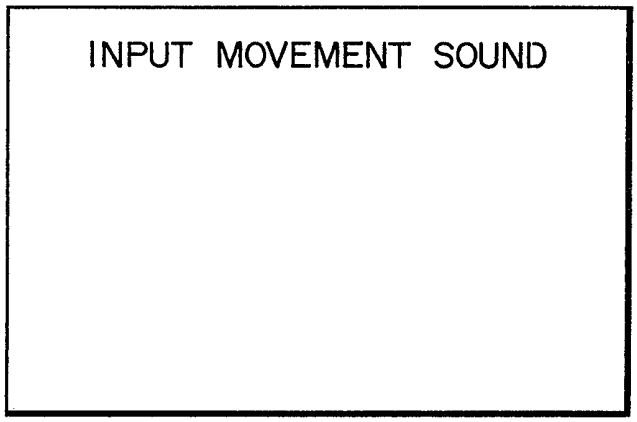


FIG. 33C

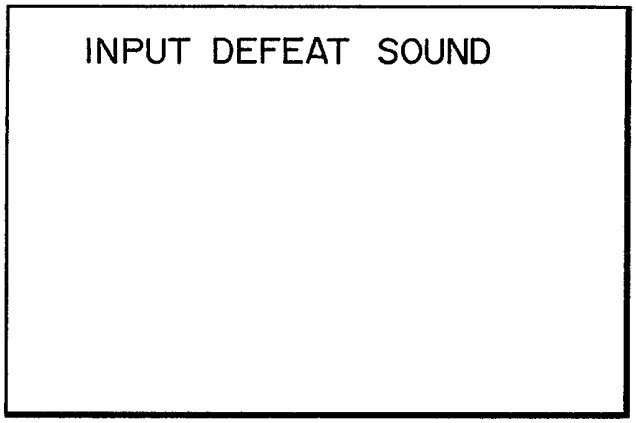


FIG. 33D

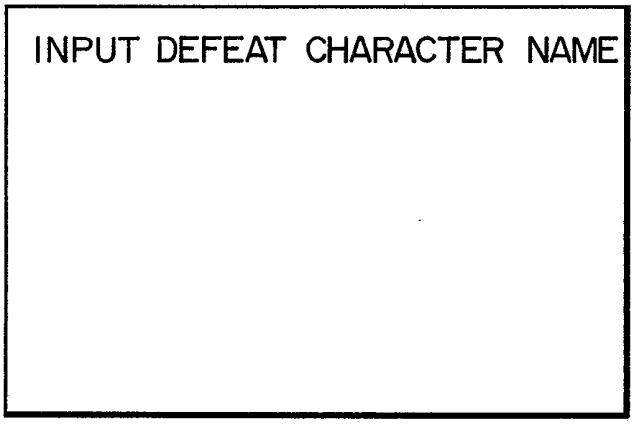


FIG. 33E

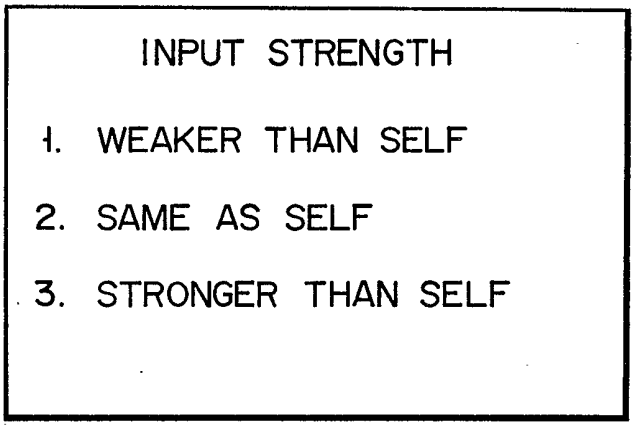


FIG. 33F

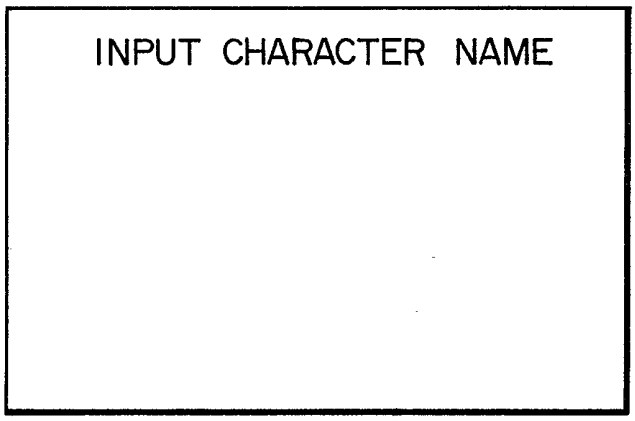


FIG. 33G

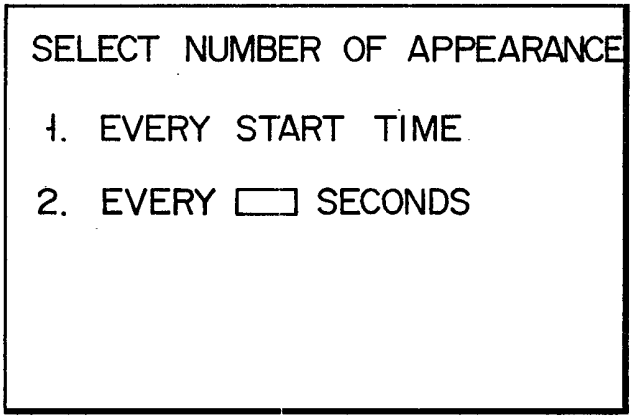
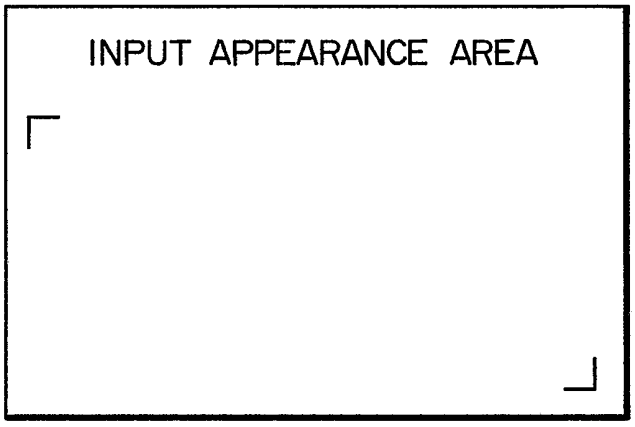


FIG. 33H



2163929

FIG. 34A

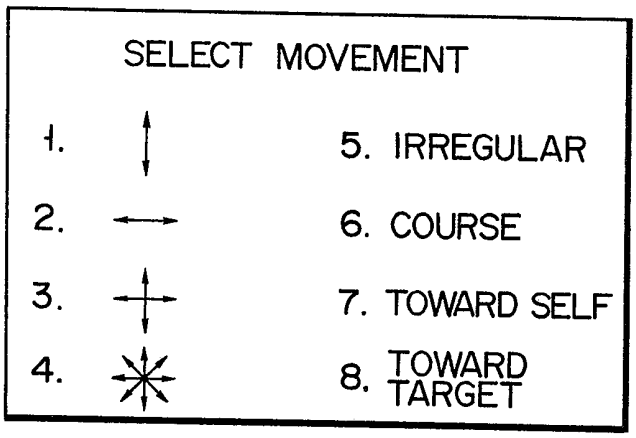


FIG. 34B

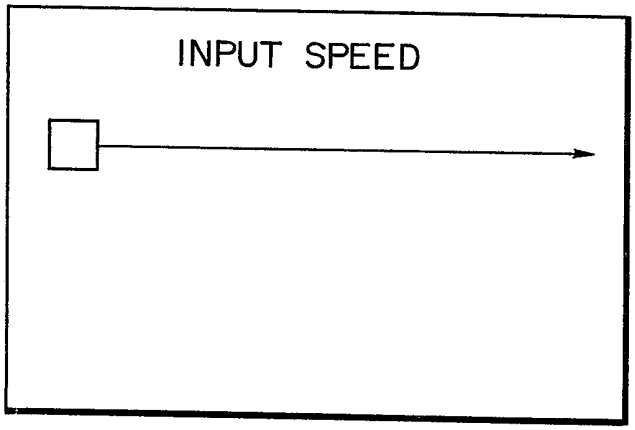


FIG. 34C

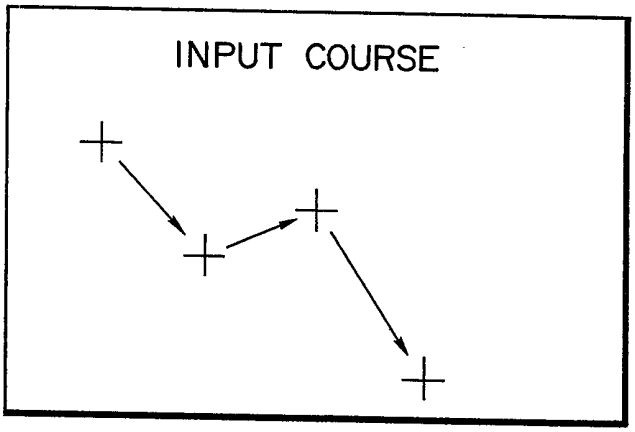
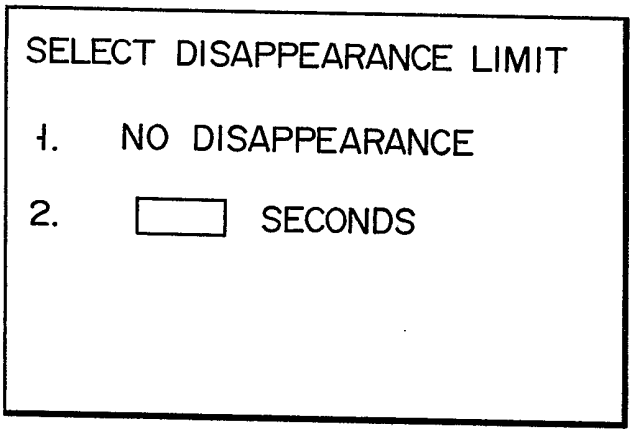


FIG. 34D



2163929

FIG. 34E

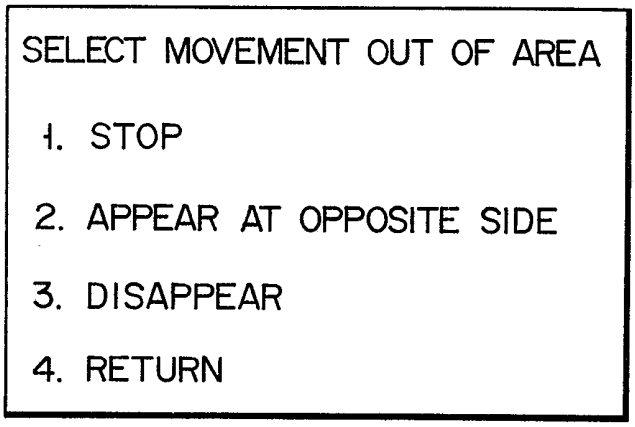


FIG. 34F

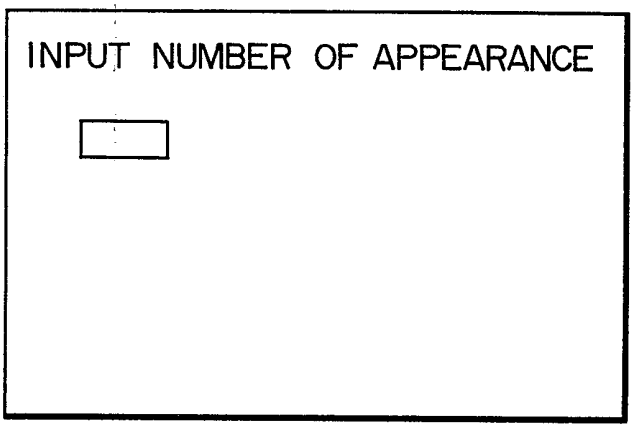


FIG. 34G

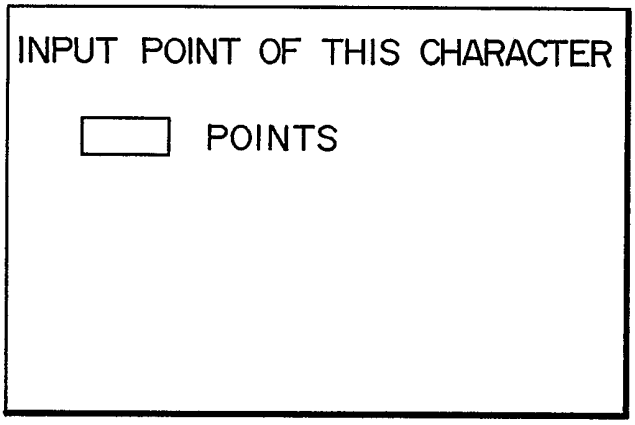


FIG. 34H

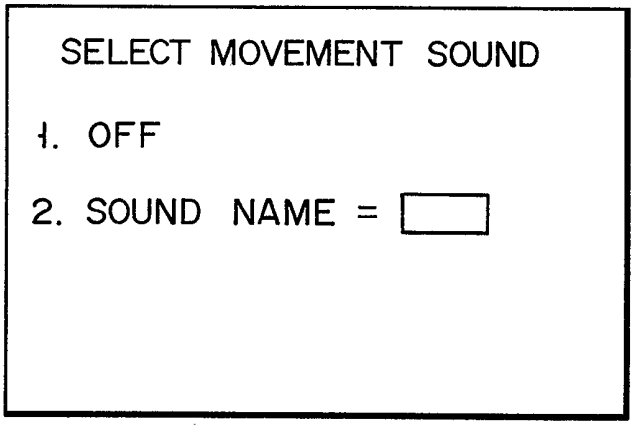


FIG. 35A

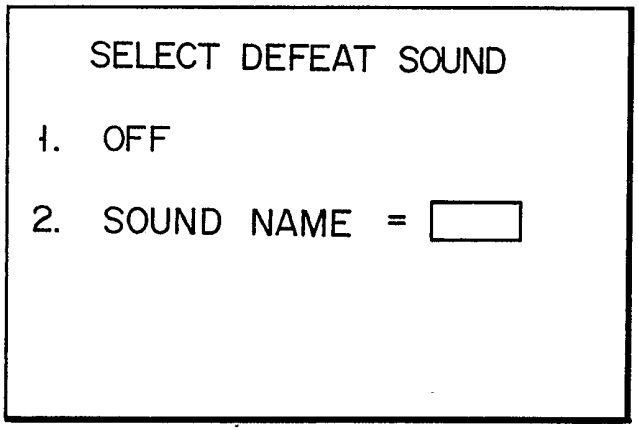


FIG. 35B

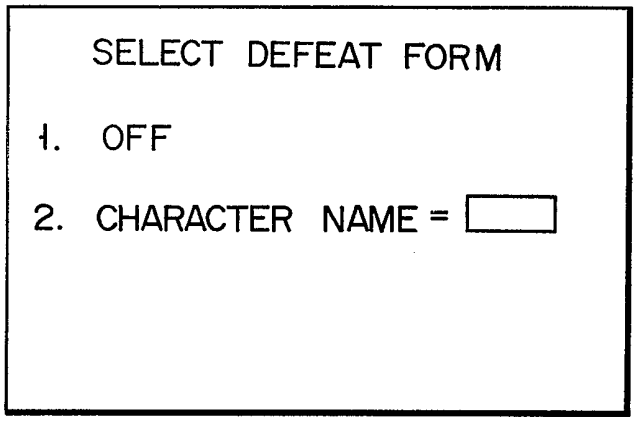


FIG. 35C

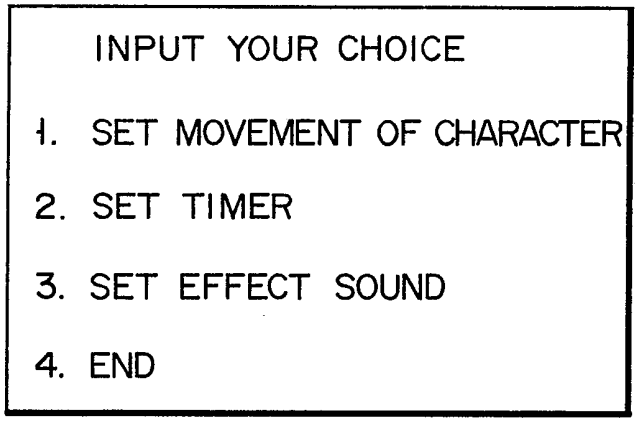


FIG. 35D

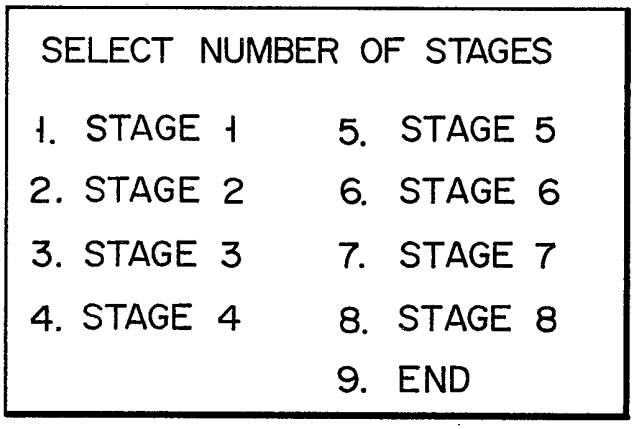


FIG. 36A

INPUT YOUR CHOICE

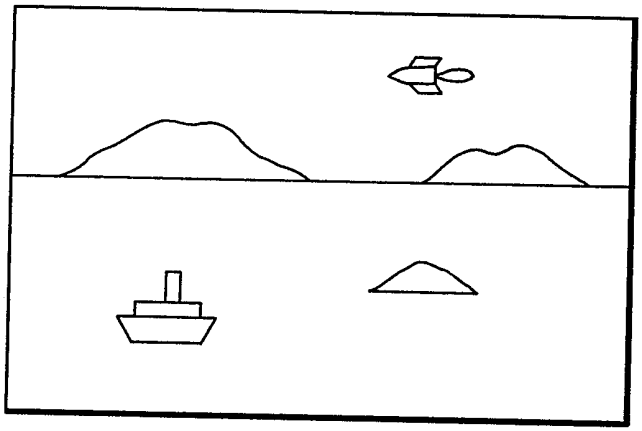
1. DRAW UP BACKGROUND IMAGE
2. DRAW UP CHARACTER
3. DRAW UP STORY
4. DRAW UP EFFECT SOUND
5. END

FIG. 36B

INPUT YOUR CHOICE

1. PERFORM GAME
2. DRAW UP GAME
3. PRESERVE GAME
4. READ IN GAME

FIG. 36C



2163929

FIG. 37

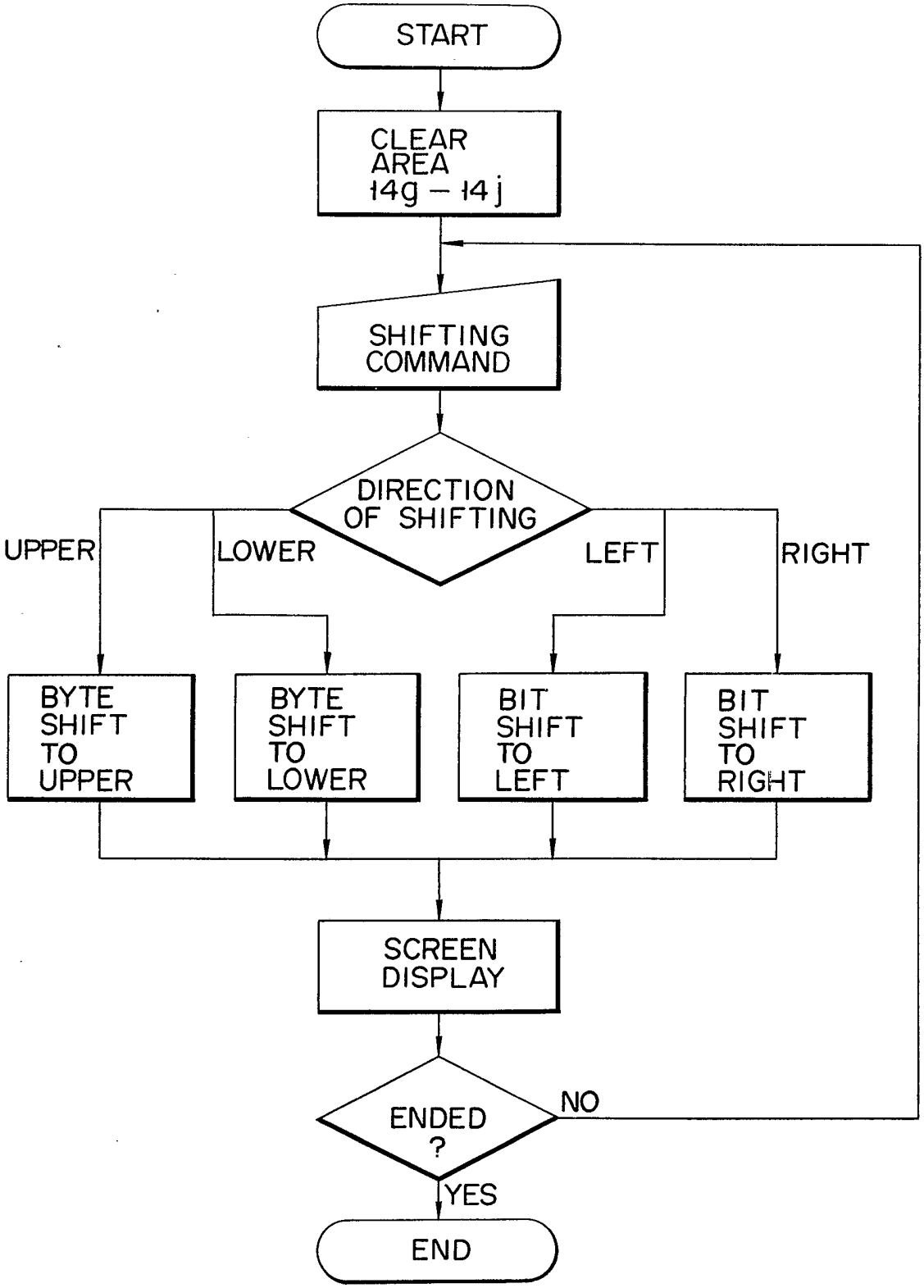


FIG. 38

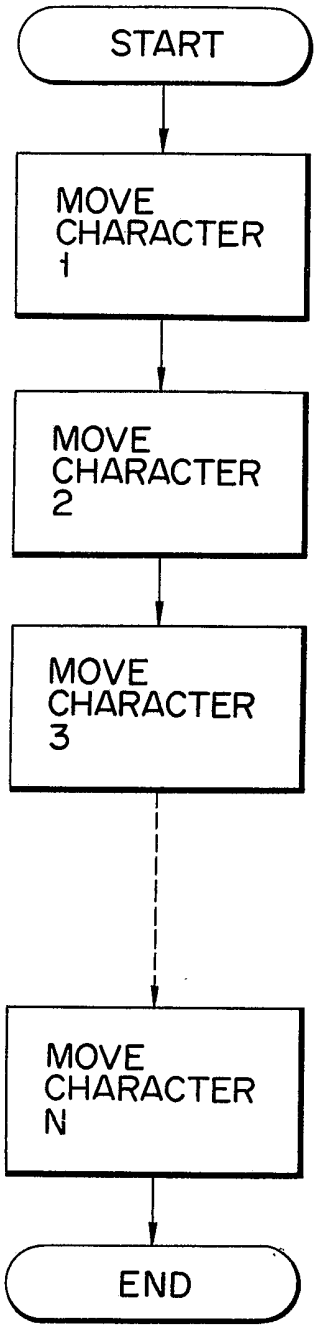

START
MOVE CHARACTER 1
MOVE CHARACTER 2
MOVE CHARACTER 3
MOVE CHARACTER N
END

FIG. 39

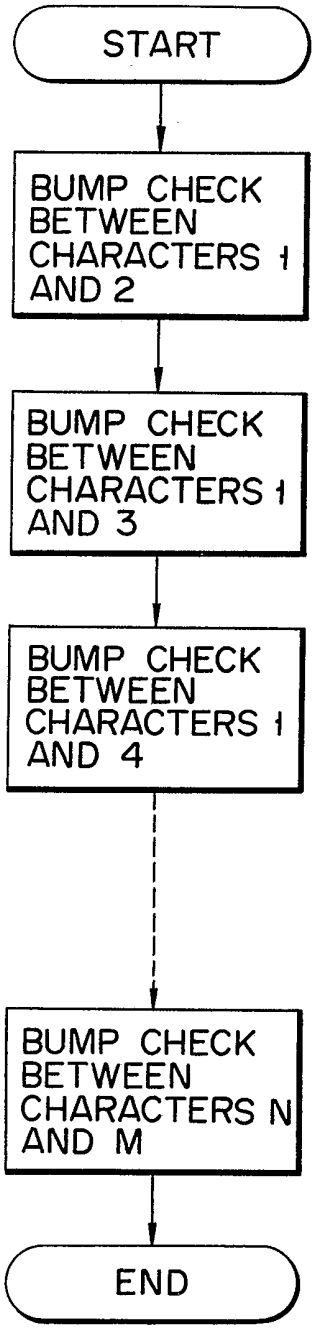

START
BUMP CHECK BETWEEN CHARACTERS 1 AND 2
BUMP CHECK BETWEEN CHARACTERS 1 AND 3
BUMP CHECK BETWEEN CHARACTERS 1 AND 4
BUMP CHECK BETWEEN CHARACTERS N AND M
END

FIG. 40

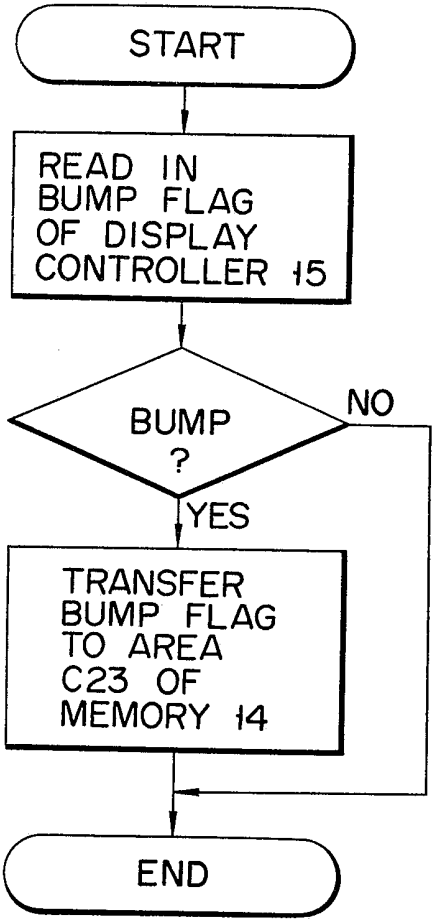
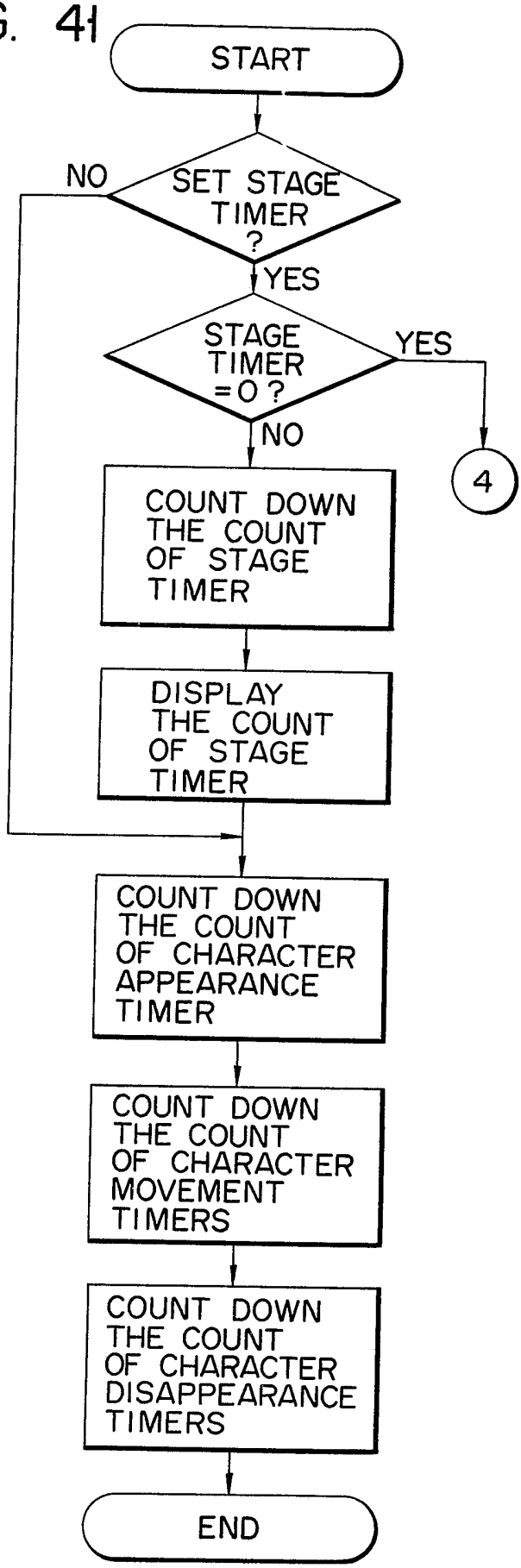
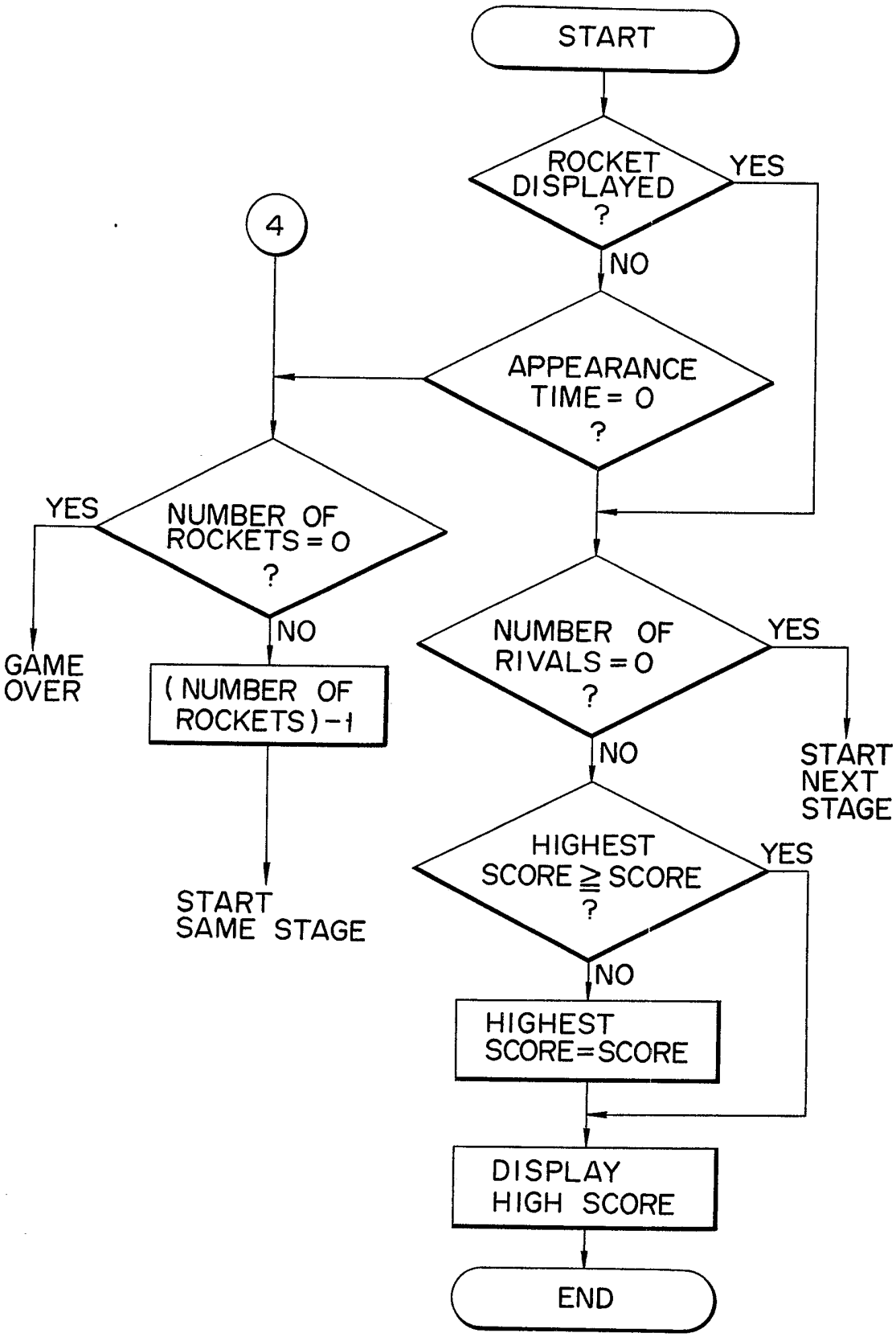

START
READ IN BUMP FLAG OF DISPLAY CONTROLLER 15
BUMP ?
NO
YES
TRANSFER BUMP FLAG TO AREA C23 OF MEMORY 14
END

FIG. 41


START
SET STAGE TIMER ?
NO
YES
STAGE TIMER =0 ?
YES
NO
4
COUNT DOWN THE COUNT OF STAGE TIMER
DISPLAY THE COUNT OF STAGE TIMER
COUNT DOWN THE COUNT OF CHARACTER APPEARANCE TIMER
COUNT DOWN THE COUNT OF CHARACTER MOVEMENT TIMERS
COUNT DOWN THE COUNT OF CHARACTER DISAPPEARANCE TIMERS
END

2163929

FIG. 42



SPECIFICATION

Electronic game apparatus

This invention relates to an electronic game apparatus using an animation display and, more particularly, to an electronic game apparatus suitable for a television game apparatus and the like.

Recently, various television game apparatuses, i.e., electronic game apparatuses, e.g., an "invader game", a maze game and the like have been developed. In such electronic game apparatuses, when a game cassette is loaded to a game performing apparatus such as a personal computer, a user can enjoy various games.

However, since such games are ready-made and specifications thereof are predetermined, they cannot be changed in accordance with the desires of the user. In general, since personal computer users are creative, they are not satisfied simply by playing ready-made games.However, it is very difficult to program a game using a computer language as BASIC, because it requires much time and effort.

It is, therefore, an object of the present invention to provide a new and improved electronic game apparatus using an animation display which a player can use to create and perform a desired game without complex programming.

According to the present invention, an electronic game apparatus using an animation display comprises: game form setting means for designating basic game algorithm data of a desired game of a predetermined form; background image setting means for setting background image data to be added to the designated basic game algorithm data; character setting means for setting character data to be added to the designated basic game algorithm data; game story setting means for setting game story data to be added to the designated basic game algorithm data; and game performing means for adding the background image data, the character data and the game story data to the designated basic game algorithm data so as to display it on the animation display, thus performing the game of the predetermined form.

These and other objects and features of the present invention can be understood by reference to the accompanying drawings, in which:

Figure 1 is a block diagram showing an overall arrangement of an electronic game apparatus according to an embodiment of the present invention;

Figure 2A and *2B* are tables for explaining the relationship between a pattern of a background image block and a color code used in this embodiment;

Figure 3 is a view showing a character attribute table 20 of Figure 1;

Figure 4 is a table for explaining an attribute code used in Figure 3;

Figure 5 is a view showing character patterns in different directions used in Figure 3;

Figure 6 is a view showing character patterns used in Figure 3;

Figure 7 is a representation showing a character definition table 21 in Figure 1;

Figure 8 is a view showing an arrangement of a character pattern generator 22 in Figure 1;

Figure 9 is a view showing an arrangement of a memory group 14 in Figure 1;

Figure 10 is a flow chart showing the overall operation of this embodiment;

Figure 11 is a flow chart showing a game drawing-up operation of this embodiment;

Figure 12 is a representation for explaining a storage content of the character attribute table 20 of Figure 3;

Figures 13A to *13C* are representations for explaining shift patterns of a character pattern of this embodiment;

Figure 14 is a representation for explaining a shift operation of the character pattern of this embodiment;

Figure 15 is a flow chart for explaining a story setting operation of this embodiment;

Figures 16A and *16B* are representations for explaining a movement direction setting operation of the character of this embodiment;

Figure 17 is a flow chart showing a movement course setting operation of the character of this embodiment;

Figure 18 is a representation showing a storage state of a table for storing movement course data of the character of this embodiment;

Figure 19 is a view showing an arrangement of a menu screen of this embodiment;

Figure 20 is a flow chart showing the overall game performing operation of this embodiment;

Figure 21 is a flow chart showing a routine operation of Figure 20;

Figure 22 is a flow chart showing a character movement operation of this embodiment;

Figure 23 is a view for explaining a character movement direction;

Figure 24 is a flow chart showing a dump check operation of this embodiment;

Figures 25A to *25E*, *Figures 26A* to *26D*, *Figures 27A* to *27C*, *Figures 28A* and *28B*, *Figures 29A* to *29J*, *Figures 30A* to *30D*, *Figures 31A* to *31D*, *Figures 32A* to *32D*, *Figures 33A* to *33H*, *Figures 34A* to *34H*, *Figures 35A* to *35D*, and *Figures 36A* to *36C* are respectively representations showing examples of menu and message displays;

Figure 37 is a flow chart showing a character shift operation of this embodiment; and

Figure 38 to *42* are flow charts showing main steps in Figures 20 and 21 in more detail.

Figure 1 is a block diagram showing the overall arrangement of the present invention. Referring to Figure 1, reference numeral 1 denotes a key input device comprising a number of keys and a joystick needed to draw up a game and to play it.A key input signal from the device 1 is supplied to an input controller 2. The controller 2 is a processor which controls the overall operation of the system in response to the key input signal, and stores algorithm data for controlling various control devices. Reference numeral 3 denotes a game draw up processor. The processor 3 starts its operation in response to an operation start instruction from the

controller 2, and supplies an end signal when the operation ends. The processor 3 is connected to a menu display control device 4 for controlling a menu display (including messages) on a CRT screen, as will be described later. The processor 3 performs game drawing-up processing set by the key input signal while controlling the menu display control device 4 to display the menu screen representing the draw up procedures for a game on the CRT screen. The menu display control device 4 can be controlled by the input controller 2. The controller 2 is connected to game algorithm memories 6 to 9 through a selector 5. The memories 6 to 9 store basic algorithm data, e.g., battle form, maze form, step form and motor race form data, and one of them is selected by a game form selection signal supplied from the selector 5. Reference numeral 10 denotes a background image processor for desirably setting a background image of the game on the CRT screen; 11, a character processor for setting desired characters; and 12, a game story processor for setting a game story to be drawn up, i.e., movement of characters, scoring, a game time, and the like. Reference numeral 13 denotes an effect sound processor for setting an effect sound used in the game to be drawn up.

The processors 10, 11, 12 and 13 and the menu display control device 4 are driven by the processor 3 and are connected to a memory group 14, and various set data are registered in the memory group 14. The processors 10 to 13 and the menu display control device 4 are also connected to a display controller 15, and current set data can be displayed on a CRT 16.The memory group 14 is also connected to the processor 3,.

The game draw up processor 3 stores algorithm data for controlling the processors 10, 11, 12 and 13, the menu display control device 4 and the controller 15. The processors 10 to 13 and the menu display control device 4 have the functions of the processor 3 and store algorithm data of corresponding parts of the game drawing-up processing executed by the processor 3. However, in view of circuit design, the processor 3 can perform the processor functions of the processors 10 to 13 and the menu display control device 4.

The display controller 15 synthesizes and displays the background image and animated pictures, and can comprise a well known processor, e.g., TMS9918A, TMS9118 etc. available from Texas Instruments, Co., Ltd. Therefore, in this specification, the display control operation is schematically described. A background image display refresh memory 17 connected to the controller 15 stores, e.g., pattern codes of 24 × 32 blocks in one frame.Each pattern code has an 8-bit configuration, and the pattern codes consist of 1st to 255th codes. Reference numeral 18 denotes a pattern generator for storing patterns corresponding to the pattern codes stored in the memory 17. Reference numeral 19 denotes a color table for storing color codes corresponding to the pattern codes. The generator 18 has addresses in one-to-one correspondence to those of the memory 17, and stores up to 255 block patterns of the 8-bit configuration corresponding to the pattern code written in the memory 17, as shown in Figure 2A. The color table 19 designates a color of the block pattern in units of lines (8 dots), as shown in Figure 2B, and stores a color code of a color of "1" (a display pattern) and that of a color of "0" (a background image of the display pattern) in 8 bits in one line.The background image can be set in units of blocks when the pattern codes of the memory 17 are rewritten, and can be set in units of dots when dots of the character patterns of the generator 19 are rewritten while fixing the correspondence between the pattern codes of the memory 17 and the block patterns of the generator 19. Note that since the memory 17 has a capacity corresponding to three frames, the background images for up to three frames can be set in units of blocks.

Reference numeral 20 denotes a character attribute table for storing attributes of characters which are displayed in motion on the background image, i.e., which appear in the game. Reference numeral 21 denotes a character definition table for storing display coordinates of characters, numbers (Nos.) of the character patterns, and color codes; and 22, a character pattern generator for storing the character patterns of the characters. Figure 3 shows an arrangement of the table 20. Referring to Figure 3, reference numeral 20b denotes an area for storing a character name; and 20a, an area for storing an attribute code for discriminating whether or not the character is a self character. The attribute code is provided for discriminating five character attributes 000 to 100, as shown in Figure 4. Referring again to Figure 3, reference numerals 20c to 20j denote areas for storing table numbers (#) of the table 21. Since up to eight types of character patterns can be set with respect to a single character name, the eight areas 20c to 20j as "upward surface", "upward back", "under surface",..., and "right back" areas are provided. For example, as shown in Figure 5, eight character patterns can be set in accordance with movement directions of the character, or as shown in Figure 6, surface and back modified patterns can be set. If the modified patterns are alternately displayed as shown in Figure 6, the character can be moved realistically. Reference numeral 20k denotes an area for storing addresses in the memory group 14 for storing various data such as strength data of the character. Note that even when eight character patterns are set with respect to a single character, the attribute and other data can be commonly used therefor.

Figure 7 shows an arrangement of the table 21. As shown in Figure 7, the table 21 defines 32 characters of table numbers #1 to #32, and stores in each table the display coordinates (x,y) in areas 21a and 21b, respectively, the number (No.) of the character pattern in the generator 22 in an area 21c, and the color code for the corresponding character in an area 21d. The color code designates a color of a dot "1" and that of a dot "0" of the character block (16 dots × 16 dots). Therefore, a single character pattern is drawn with a single color (two colors including a background image color). The numbers (No.) written in the areas 20c to 20j of the

table 20 correspond to the table numbers #1 to #32. When the character name and one of upward surface data,... left back data are designated, the table number of the table 21 is designated, and the pattern number of the generator 22 is then designated. In this embodiment, the table numbers #1 to #32 of the table 21 have one-to-one correspondence to the pattern numbers No. 1 to No. 32. Therefore, the pattern code is written fixed in the area 21c of the corresponding table number, e.g., the pattern code No. 1 is written in the area 21c of the table number #, and the pattern code No. 2 is written in the area 21c of the table number #2. The display operation control of the characters can be performed by controlling the data in the table 21. Accordingly, when the coordinate data in the areas 21a and 21b are sequentially rewritten, the character pattern of the pattern number designated by the area 21c is displayed in motion. When the y coordinate of the area 21b is set to be out of the area, the character is not displayed.

Figure 8 shows an arrangement of the character pattern generator 22. A single character is configurated by 16 dots × 16 dots, and 32 character patterns No. 1 to No. 32 can be set. Therefore, when a single character pattern is set with respect to a single character name, 32 character names can be set.When 8 character patterns are set with respect to a single character name, 4 character names can be set.

The controller 15 displays the background image by means of the memory 17, the generator 18, and the table 19, and displays characters (animation) by means of the tables 20 and 21 and the generator 22.In this case, a priority is #1 > #2 > ... > #32 > (the block pattern of the background image) when the displays overlap each other.When the character patterns overlap each other, a dump flag "1" is set in the controller 15.The controller 15 is connected to the memory group 14 so that the storage contents of the memory 17, the generator 18, tables 19 to 21 and the generator 22 connected to the controller 15 can be transferred to the memory group 14.

Reference numeral 23 denotes a game performing device. The device 23 is a processor which accesses one of the memories 6 to 9 selected by the controller 2 through the selector 5, and executes a selected game algorithm in accordance with various data written in the memory group 14 by the device 3. Upon execution, the device 23 drives the controller 15 for displaying, and simultaneously drives an effect sound generator 25 so as to produce an effect sound from a loudspeaker 26. The device 23 is operated upon reception of an interrupt signal generated from the controller 15 at 16-msec intervals. The generator 25, connected to the processor 13, stores sound data set by the processor 13, and generates a sound corresponding to the sound data designated by the device 23. The generator 25 also stores a plurality of effect sound data which have been prepared in advance and are provided with predetermined sound names.The processor 13 provides a name to the sound data, and designates the provided name or preset name, thus generating a desired sound.

Figure 9 shows a main arrangement of the memory group 14, and meanings of data stored in respective areas will be described after description of an operation thereof. Note that referring to Figure 9, an area A stores data associated with an entire game, an area B stores data associated with an entire stage, and other areas store data associated with individual characters. Particularly, the same attribute written in the area 20a of Figure 3 is written in an area C22. If either the area 20a or 20c is accessed, the same attribute data can be obtained. A condition flag area C23 stores a code indicating a display/non-display state and a direction of a character. A condition flag "0" indicates a non-display state, and condition flags "1" to "8" indicate a display state. The direction of the character is indicated by a direction number of Figure 16A.The flag is updated by step S79 of Figure 22 during an operation. An area C10 stores a parent character number (#).

Although not shown in Figure 1, various game data registered in the memory group 14 can be stored in an external storage device such as a floppy disk or a magnetic tape together with the contents of the generator 18, the tables 19, 20 and 21 and the generators 22 and 25, and conversely, the storage data can be loaded into the apparatus.

<Operation of Embodiment>

The operation of the apparatus of this embodiment will be described hereinafter.Figure 10 is a flow chart showing the overall operation of the apparatus executed by the input controller 2. When the apparatus is started by turning on a power switch (not shown), an initial menu screen is displayed in step S1. The menu screen is displayed on the CRT 16 in a manner such that the controller 2 accesses the menu display control device 4 and reads out the initial menu screen, as shown in Figure 25A (the following menu screens are displayed in the same manner). The apparatus has "game drawing-up", "game performing", "game preserving" and "game reading in" functions, and the initial menu screen is continuously displayed until the user selects the desired function. When one of the functions is selected by a cursor key of the device 1 in step S2, it is judged in step S3 which function is selected, and the flow jumps to a corresponding processing.Step S4 is the step of drawing up the game, and will be described later in detail. Step S5 is the step of performing the game, and will also be described later in detail. Step S6 is the step of preserving the drawn up game in an external storage device. However, since it is a state-of-art technique and not directly related to the present invention, a detailed description of this step is omitted. Step S7 is the step of reading in the preserved game, and a detailed description thereof is omitted for the same reason as in step S6.

The game drawing up operation in step S4 will first be described in detail hereinafter.

<Game Drawing Up Operation>

Figure 11 is a general flow chart of the game

drawing up operation. When the controller 2 supplies an operation start signal to the processor 3, the processor 3 starts the flow. In step S11, the menu display is performed for selecting a game form, as shown in Figure 25B.In this step, the menu is displayed for selecting one of a battle form game (game I), a maze form game (game II), a step form game (game III) and a motor race form game (game IV). When one of the game forms is selected by the cursor key of the device 1 (the following selections are made in the same manner as described above), the selected game form is registered in an area Al of the memory group 14 in step S13. In this case, a description will be made assuming that the battle form game is selected and the battle form game is created. Note that the battle form game is also referred to as an invader form game, in which a self character (one's own) and rival characters freely run about the screen, and the self character shoots down the rival characters while avoiding their attacks. In step S14, a menu display shown in Figure 25C is made to select an item to be created next. In order to draw up the game, "background setting", "character setting", "effect sound setting" and "story setting" operations must be performed, and an instruction indicating each item to be set next is awaited. One of the operations is selected by the device 1 in step S15, and it is judged in step S16 which item is selected. Then, the flow advances to one of steps S17 to S20 in accordance with the selection. Step S17 is the step of setting the background image, and a setting start instruction is supplied to the processor 10.

Background Image Setting Operation

The processor 10 requests a key input from the device 1 while accessing the menu display control device 4 so as to sequentially display the menu screens, and draws up the background image.In this case, a screen mode is selected from a "1 frame only" mode, "3-frame switching" mode and "scrolling" mode, and the selected mode is written in an area A2 of the memory group 14. Since the memory 17 has a capacity corresponding to three frames, the screen can be switched for up to three frames for each stage, and can be scrolled for three frames. In the scrolling mode, the screen can be scrolled automatically or in accordance with the movement of the self character. The type of scrolling mode is also registered in the area A2 of the memory group 14. Then, the background image is actually drawn. When the 1-frame only mode is selected, the background image is drawn in units of dots. When the 3-frame switching mode or the scrolling mode is selected, the background image is drawn in units of blocks. Thus, the background image is drawn in a desired color upon moving a cursor. This graphic technique is well known to those skilled in the art, and a detailed description thereof will be omitted. When the background image is completed, the processor 10 supplies an end signal to the processor 3. Then, the flow returns to step S14, and the menu display shown in Figure 25C is made so as to await the next key input.

When the character setting operation is selected, the flow advances to step S18 through steps S15 and S16.

Character Setting Operation

In step S18, the processor 3 supplies a character setting start signal to the processor 11. The processor 11 accesses the menu display control device 4 so as to display a menu for setting a character, as shown in Figures 26A and 26B (where "1. DRAW UP CHARACTER" is selected from the menu of Figures 26A). In accordance with a menu for selecting a character to be drawn up, "self", "arms of self", "rival", "arms of rival" and "defeat form" characters are drawn up. Although all the character need not always be formed, at least the self and rival characters are required. When "SELF" is selected from the menu of Figure 26B, a message "INPUT CHARACTER NAME" shown in Figure 26C is displayed. Thus, "ROCKET" is inputted from the device 1, as shown in Figure 26C. The attribute code indicating the self character is written in the area 20a of the table 20, and the character name "ROCKET" is written in the area 20b thereof. After the character is inputted, when the return key of the device 1 is depressed, a message asking whether or not a direction of the character is changed is displayed, and is selected, as needed. In this case, assume that "2. YES" is selected.Then, a message "DRAW UP CHARACTER" and a rectangular frame of 16 dots × 16 dots are displayed, as shown in Figure 27A. A desired character pattern of 16 dots × 16 dots is drawn up in units of dots in the frame using the cursor key and a color key of the device 1, e.g., as shown in Figure 27B. When one character is completed by depressing the return key, the next rectangular frame and an arrow indicating a direction thereof are displayed, as shown in Figure 27C, and up to eight characters can be drawn up with respect to one character name. Of course, any number of characters can be formed up to eight. As shown in Figure 5, when eight character patterns are formed, the table numbers #1 to #8 of the table 21 are written in the areas 20c to 20j of the table 20. The table numbers are automatically assigned. The pattern numbers of the generator 22 are written in the corresponding areas 21c. In other words, the pattern numbers No. 1 to No. 8 are written in the areas 21c of the table 21 for the table numbers #1 to #8. The color code (color for a character pattern and that of a background image) for setting the character pattern is written in the area 21d. The character pattern is written in areas No. 1 to No. 8 of the generator 22 shown in Figure 8. On the other hand, a start address of an area B9 of the memory group 14 is written in the area 20k of the table 20.

As shown in Figure 12, the table numbers of the table 21 are sequentially written in the table 20 from the table number #1 in the order of forming the character patterns. Start addresses of character data areas (B9, B10,... B13) of the memory group 14 which are prepared in advance are written in the corresponding areas 20k of the table 20 in the order of forming the character patterns.

An operation for shifting the character pattern within a character block will be described hereinafter. Assume that a character pattern as shown in Figure 13A is drawn up in advance. Since this pattern has a poor appearance, it may be desired to shift the entire pattern downward and to add a pattern to an upper portion thereof. However, it is very cumbersome to form the pattern from the beginning. For this reason, in this embodiment, a pattern shift function in the character block is provided. As shown in Figure 14, a character buffer area 14f and save areas 14g to 14j are provided in a work area WA of the memory group 14. When pattern shift processing is to be performed, a target character pattern is transferred from the generator 22 into the area 14f of the memory group 14 through the controller 15. Referring to Figure 14, although each area is represented by 4 × 4 blocks for the sake of simplicity, each area has 16 bits × 16 bits capacity in practice. When a downward shift instruction is provided from the device 1, each 16-bit row is shifted downward by one bit, and the lowest row shifted outside the area 14f is stored in the area 14i. When a right shift instruction is provided, each 16-bit column is shifted to the right by one bit, and the rightmost column shifted outside the area 14f is stored in the area 14h. The same operation is conducted when upper and left shift instructions are provided. When the pattern shown in Figure 13A which is displayed by a predetermined operation (to be described later) is shifted by 3 dots by using the cursor key, the resultant pattern is as shown in Figure 13B. Then, an additional pattern is formed on an upper portion of the pattern, and a character pattern having good appearance can be formed, as shown in Figure 13C. Thereafter, the content in the area 14f is transferred to the generator 22. Figure 37 is a flow chart showing the above operation which is executed under the control of the processor 11. Note that an actual shift operation is performed in such a manner that a user displays a menu screen shown in Figure 26A, selects an item "2. CORRECT CHARACTER" to display a menu screen shown in Figure 28A, and selects an item "2. SHIFT". In this case, a character name of a character to be shifted must be inputted in accordance with a message "INPUT CHARACTER NAME" shown in Figure 28B.

When the character setting operation is completed, the processor 11 supplies an end signal to the processor 3, and the processor 3 executes step S14 of Figure 11 again.

Effect Sound Setting Operation

Step S20 is the step of setting an effect sound prior to a story setting operation to be described later.When an item "4. DRAW UP EFFECT SOUND" is selected from the menu screen shown in Figure 25C, the processor 3 supplies an effect sound setting start signal to the process 13. The processor 13 accesses the menu display control device 4 so as to display a menu screen including a message "INPUT YOUR SELECTION", as shown in Figure 29A. Assume that an item "1. CREATE SOUND" is selected. Then, a message "INPUT SOUND NAME" shown in Figure 29B is displayed, and, for example, "SOUND 1" is inputted from the device 1. Thus, since a music sheet is displayed together with various menus and messages required for setting a sound, an arbitrary sound can be set on the music sheet in accordance with the menus and messages by the device 1. A detailed operation will be described later. The sound data is stored in the generator 25 together with the sound name.

When the effect sound setting operation is completed, the processor 13 supplies an end signal to the processor 3.

An actual sound setting process after inputting the sound name "SOUND 1" will be described hereinafter. As shown in Figure 29C, a menu screen representing types of sounds to be set is displayed. Assuming that an item "1. MELODY" is selected, a menu screen for selecting the way of drawing up sound is displayed, as shown in Figure 29D. In this case, if an item "1. SAMPLE" is selected, a menu screen representing music names prepared as samples in advance is displayed, as shown in Figure 29E. On the other hand, if an item "2. CREATE SOUND" is selected from the menu of Figure 29D, a music sheet on which notes are provided at given positions is displayed together with information "MUSIC PIECE", "TEMPO", "REPEAT" and "CHORD" necessary for creating sound. Appropriate information can be selected by using cursor keys and and the return key. Particularly, when an item "1. MUSIC PIECE" is selected, horizontal positions of notes are designated by using cursor keys and , and pitches thereof are designated by using cursor keys and . A rest can be obtained by depressing a rest key.An item "REPEAT" is selected when the melody is to be repeated and an item "CHORD" is selected when a chord is provided to the melody. Tempo is gradually increased in an order of "1" to "9".

When an item "2. CORRECT SOUND" is selected from the menu screen of Figure 29A, a menu screen related to designation of a sound name to be corrected is displayed, as shown in Figure 29G. Assuming that an item ' 2. SOUND 2" is selected, a menu screen shown in Figure 29H is displayed so as to inquire which information for the sound, set as the sound name "SOUND 2", is to be corrected is displayed. Substantially the same operation as that for creating sound shown in Figure 29F is performed for the following correction operation. After completing the correction, when an item "6. END" is selected, the menu screen shown in Figure 29A is displayed again, and when an item "4. END" is selected therefrom, the menu screen shown in Figure 25C is displayed again and the control flow returns to the routine for drawing up the game.

Note that when an item "2. EFFECT SOUND" is selected from the menu screen shown in Figure 29C, a menu screen representing various effect sounds prepared in advance is displayed, as shown in Figure 29I. When an item "1. MOVEMENT SOUND" is selected, a menu screen representing five variations of movement sounds prepared in advance is displayed, as shown in Figure 29J. If any of these variations is selected, the

selected sound is registered, and the display returns to the menu screen for drawing up the game shown in Figure 25C through the menu screen shown in Figure 29A.

Story Setting Operation

Step S19 is the step of setting a story necessary for the game to be drawn up. When an item "3. DRAW UP STORY" is selected from the menu screen shown in Figure 25C, the processor 3 supplies a story setting start signal to the processor 12. Upon reception of this signal, the processor 12 executes the flow shown in Figure 15.

In step S21, a start sound for the game to be drawn up is set. In practice, when the display 4 is accessed, a message "INPUT START SOUND" shown in Figure 30A is displayed. Thus, a sound name of the start sound, e.g., "SOUND 1", as shown in Figure 30A, is selected from the effect sounds registered in the generator 25 in advance by using the device 1.Then, the sound name of the selected sound is registered in an area A3 of the memory group 14. For the sake of simplicity, in Figure 15, the above-mentioned and following operations will be summarized as in step S21 and in processing following it. Similarly in step S22, a game over sound is set so as to correspond to a message "INPUT GAME OVER SOUND" shown in Figure 30B, and is registered in an area A4 of the memory group 14. In the step S23, the number of self characters is set in accordance with a message "INPUT NUMBER OF SELF" shown in Figure 30C. Then, the number of self characters is registered in an area A5 as an initial value.Note that a current value of the number of self characters (to be updated upon execution of the game) is registered in an area A6. It is checked in step S24 whether or not the number of self characters is increased in accordance with an increase in score so as to correspond to a message "INCREASE NUMBER OF ROCKET (SELF)?". If YES in step S24, i.e., if an item "1. YES" is selected in Figure 30D, a score at which the number of self characters is increased, e.g., 1,000 points is set in accordance with a message "INPUT INCREASE SCORE", and is registered in an area A7, in step S25. In step S26, stages are determined.In this embodiment, the stages can be set up to eight, and when, for example, stage 1 is selected, the flow advances to step S27, and otherwise, the flow advances to an other processing.

In step S27, it is judged what is to be set in stage 1 from a menu screen shown in Figure 31B. Although "MOVEMENT OF CHARACTER", "TIMER" and "EFFECT SOUND" must be set for each stage, a setting operation of "TIMER" and "EFFECT SOUND" will be omitted, and only a setting operation of "MOVEMENT OF CHARACTER" will be described. Note that in "TIMER", game time conditions are set, wherein the remaining time until the game is over and the score is increased or not increased in response to the remaining time are processed. In "EFFECT SOUND", a stage clear sound, a gaming sound and the like are set in the same manner as in sound setting operation. The data set by such processing is registered in a corresponding area of areas B of the memory group 14. Note that an area B1 stores a timer initial value; B2, a current timer value; B3, a timer score condition; B4, a stage clear sound; B5, a gaming sound; B6, scrolling start coordinates; B7, the number of characters; and B8, the total number of rival characters.

If the first item is selected from the menu screen in step S27, the character to be moved is selected from the characters "SELF", "RIVAL", "ARMS OF SELF" and "ARMS OF RIVAL" in step S28. In this case, characters "SELF" and "RIVAL" are exemplified.When an item "SELF" is selected in step S28, the flow advances to step S29. In step S29, the movement area of the self character is set. In this case, as shown in Figure 32A, an upper mark "┌" is displayed at the upper left corner on the screen and a lower mark "┘" is displayed at the lower right corner on the screen. Thus, by moving these marks using cursor keys, the movement area of the self character is defined as a rectangular area enclosed by the marks "┌" and "┘". Leftmost and rightmost values XL and XR of the x coordinate and uppermost and lowermost values YU and YD of the y coordinate in the defined area are registered in an area C8 of the memory group 14. In step S30, parameters associated with movement of the character are set in accordance with a menu screen shown in Figure 32B. In this case, the movement direction of the character, i.e. an up-and-down direction, a left-and-right direction, up-and-down and left-and-right directions, or eight directions is determined. In this case, the parameter setting operation associated with the movement of the character is performed in such a manner that a user selects from eight styles defining eight directions as directions 1 to 8, as shown in Figure 16A, i.e., codes 1 to 8 used for setting "1" in a bit corresponding to a movable direction, as shown in Figure 16B, and the selected code is registered in an area C19. Note that in the following steps, the parameter setting operation is to select one of some styles prepared in advance and to register a code corresponding to the selected style in the memory group 14. In step S31, in correspondence with a message "INPUT SPEED" shown in Figure 32C, a speed for moving the character is determined. That is, a dummy character appears on the screen, and is repeatedly moved from the left to the right on the screen. When the ⊡ key of the device 1 is depressed, a time required for moving the character by one step is shortened, i.e., the speed is increased. In contrast to this, when the ⊡ key is depressed, the speed is decreased. Furthermore, when the ⊡ key is depressed, the number of dots for moving the character by one step is increased, i.e., the speed is increased. In contrast to this, when the ⊡ key is depressed, the speed is decreased. An optimum speed can be set by appropriately depressing these keys. In the parameters of the speed, an initial value of time (movement timer) required for moving the character by one step is registered in an area C14 of the memory group 14, a current value is registered in an area C15, and a dot number for one step is registered in

an area C21. In step S32, processing for the character outside the area set in step S29 is performed in accordance with a menu screen shown in Figure 32D. In this processing, it is determined whether to stop the character, make it disappear or reappear at an opposite side of the screen, or whether to move the character in a reverse direction (return the character), and the determined parameter is registered in an area C16. In step S33, the character name is inputted in accordance with a message "INPUT THE CHARACTER NAME" shown in Figure 33A. As described above, this character name must be one set by the processor 11. The present character name is registered in an area C1. In steps S34 and S35, in correspondence with messages "INPUT MOVEMENT SOUND" and "INPUT DEFEAT SOUND" shown in Figures 33B and 33C, a movement sound and a defeat sound for the self character are selected from the effect sounds set in the generator 25, and the selected sounds are registered in areas C6 and C7, respectively. In step S36, in correspondence with a message "INPUT DEFEAT CHARACTER NAME", a character name for a defeat character set in the generator 22 is selected, and is registered in an area C2.When the movement of the self character is set, the flow returns to step S27, and the item "SET MOVEMENT OF CHARACTER" is selected from the menu screen shown in Figure 31C. Then, an item "RIVAL" is selected from the menu screen shown in Figure 31D in step S28, and the flow advances to step S37.

In step S37, a strength of a character (RIVAL) is set in accordance with a menu screen shown in Figure 33E. Three strengths, "0", "1" and "2" are provided, becoming weaker in the order of "2" → "1" → "0", and the strength of the self character is fixed at "1". When two characters collide, the weaker one explodes or disappears. When characters having the same strength collide, both disappear. A strength code is set in an area C13. Note that in a memory map of Figure 9, the areas C1 to C23 used for storing data of the self character are also used for storing data of the rival character. However, in practice, the data of the rival character is registered in developed portions of the area B10 (not shown) for storing movement data of a second character. In step S38, an operation corresponding to a message "INPUT CHARACTER NAME" is performed, and in step S39, appearance conditions of the character are determined. Note that the appearance conditions include appearance timing of a character for every start time or for every certain number of seconds, as shown in Figure 33G, and an appearance area shown in Figure 33H. An initial value of the appearance timing is registered in an area C11 and a current value thereof is registered in an area C12. Steps S40 and S41 are the same as steps S30 and S31. That is, in step S40, "6. COURSE" is selected from the menu screen as shown in Figure 34A. In step S41, a setting similar to that of step S31 is performed in response to a message "SET MOVEMENT SPEED". In step S42, a course along which the character is moved is determined. In this embodiment, when a start point and a plurality of junctions are designated by using the cursor keys and the space key, the character is displayed from the start point via the junctions, and a desired course can be easily set. Figure 17 shows step S42 in detail.

Referring to Figure 17, in step S421, in correspondence with a message "INPUT COURSE" shown in Figure 34C, a "+" mark is displayed as a cursor at a predetermined point on the screen. In step S422, when the cursor is moved by cursor keys, the flow returns from step S423 to S421, and a cursor display position is shifted. When a set key is depressed, the flow advances from step S423 to S424, and currently displayed cursor coordinates are registered as a start point in an area C20 of the memory group 14. In the area C20, as shown in Figure 18 in more detail, the coordinates of the start point are registered in area C20a. In step S425, the character having the character name designated in step S38 is read out from the generator 22, and is displayed at the start point coordinates. In other words, the area 20b of the table 20 is accessed with reference to the character name registered in the area C1 of the memory group 14, the table 21 is accessed in accordance with the content of the area 20c of the table in which the accessed name is written, and the current cursor address is written in the areas 21a and 21b of the table 21.Therefore, the generator 22 is accessed in accordance with the content of the area 21c of the table 21, and the character pattern having the corresponding pattern No. is displayed at the cursor address. In step S426, a variable N is set to be "1", and in step S427, the cursor is moved by using the cursor keys. Then, the key operation is judged in step S428 and the next "+" mark cursor is displayed at a current position, in step S429. In this state, when the set key, e.g., the space key, is depressed in step S427, the key operation is judged in step S428, the flow advances to step S42A, and this cursor address is registered in an area C20b of the area C20 of the memory group 14 as the Nth junction. In this case, the cursor coordinates are not directly registered, but coordinates with reference to start point coordinates are registered. In step S42B, the character pattern is displayed so as to be moved from the start point to this junction at the predetermined speed set in step S41 (Figure 15). That is, if the contents of the table 21 are sequentially changed, a display position of the character is automatically changed. Therefore, the movement course and speed can be visually recognized. In step S42C, the variable N is incremented by one, and then the next key input is awaited. When the next junction is determined by moving the cursor, the coordinates of the second junction are registered in the area C20c, as shown in Figure 18, in the same manner as described above, and in step S42B, the character pattern is displayed so as to be moved from the start point to the second junction via the first junction. At this time, if a pattern key, e.g., the return key, is depressed, the course is determined, and the flow ends.In addition to the above-mentioned course setting operation of the character, the rival character is automatically moved toward the self charac-

ter irrespective of the position of the self character, a target point is predetermined and the rival character is moved toward the target point, or the arms of the rival character are shifted to a direction to which the corresponding character directs, as shown in the menu screen in Figure 34A.

Referring again to Figure 15, in step S43, in accordance with the menu screen shown in Figure 34D, a disappearance limit from appearance to the disappearance of the character is set, and is registered in an area C4 of the memory group 14. An area C5 stores a current value of the disappearance limit. Of course, if the character is set not to disappear, this operation can be omitted. In step S44, the same out-of-area processing as in step S32 is performed in accordance with the menu screen shown in Figure 34E, and in step S45, the number of appearances of the character is set in correspondence to the message shown in Figure 34F. Then, an initial value of the number of appearances is registered in an area C14, and a current value thereof is registered in an area C18. In step S46, a point provided with the corresponding character, i.e., a point when the self character shoots down a character is set, and is registered in an area C3. Steps S47, S48 and S49 are substantially the same as steps S34, S35 and S36 described above. That is, in steps S47, S48 and S49, the movement sound, the defeat sound and the defeat form are set in accordance with menu screens shown in Figures 34H, 35A and 35B, respectively.

In this manner, the setting operation of "self" and "rival" characters is completed.The same operation is performed for setting "arms of self" and "arms of rival", even if a plurality of types of rival characters are provided.

Thus, stages 2 and 3 are set in the same manner as described above.

Note that when the return key is depressed after step S49, the display returns to the menu screen shown in Figure 35C.In this case, if an item "4. END" is selected, the display returns to the menu screen shown in Figure 35D.Then, if an item "9. END" is selected, the display returns to the menu screen shown in Figure 36A. If an item "5. END" is selected, the display returns to the menu screen shown in Figure 36B. If an item "1. PERFORM GAME" is selected, the game is started as will be described later, and the background image and the characters are displayed in accordance with the preset movements on the CRT 16, as shown in Figure 36C.

<Menu Screen Display Operation>

The operation of the menu display control device 4 will be described hereinafter. As described above, during the game drawing up process of the present invention, various menu screens (including messages) are always displayed, and a desired item is selected from these menus so as to register parameters of the game to be drawn up. The menu screen data is prepared in the menu display control device 4, and is supplied to the controller 15 upon reception of requests from the processors so as to be displayed. The menu screen data is divided into blocks such as "initial menu M1", "game draw up menu M2", "game type setting block M3", "background setting block M4", "character setting block M5", "effect sound setting block M6", and "story setting block M7", and in each block, a plurality of menu screens are connected.A menu pointer for designating an address of each menu screen data and seven block pointers for storing start addresses of the blocks are prepared.

Referring to Figure 19, in the initial menu M1, four items "DRAW UP GAME", "PERFORM GAME", "PRESERVE GAME" and "READ IN GAME", are represented, as shown in the flow chart of Figure 10 so as to wait for the selection. When the item "DRAW UP GAME" is selected, the menu screen is changed to the game draw up menu M2, as indicated by an arrow Ma, and otherwise, after displaying other menu screens, the initial menu screen M1 is finally displayed. Since Figure 19 is provided for explaining the relationship of menu screens, the number of arrows does not coincide with the number of actual branch menu screens. Other menu screens are shown in the same manner as described above. The story setting block M7 is developed in units of menu screens.However, since these menu screens are shown for the purpose of description, the number thereof does not coincide with the actual number of the menu screens. When the story setting item is selected from the story setting block M7, a first menu of the block M7 is displayed.In the first menu, for example, a message "INPUT START SOUND" as shown in Figure 30A is displayed. When the start sound setting operation is completed, a second menu is displayed. Since an alternative selection is made in the second menu, when one item is selected, a third menu is displayed, and when the other item is selected, a fourth menu is displayed. In a fifth menu, several menus are selected, and the display returns to the fifth menu every time each processing ends. When an end item is selected in the fifth menu, the display returns to the game draw up menu M2. Figure 19 can be referred to in this manner. The menu screens are advanced by updating a menu pointer controlled by each processor. The device 1 comprises a menu back key with which the content of the menu pointer is decremented. Therefore, when the fourth menu is displayed and the menu back key is depressed, the display returns to the second menu, and when further depressed, the display returns to the game draw up menu M2. In this case, since the control flow of each processor (e.g., the game story processor 12) returns to the step of displaying a menu screen, the setting operation for each menu can be restarted. On the other hand, the device 1 comprises a menu block back key with which the block address immediately preceding the current one is set in the menu pointer, and the start menu screen of the block immediately preceding the block including the current display menu can be displayed.

<Game Performing Operation>

The game performing operation will be de-

scribed next. In order to perform the game, when the controller 2 executes step S1 for displaying the initial menu in the flow of Figure 10, i.e., when the menu screen shown in Figure 25A or 36B is displayed, the item "PERFORM GAME" is selected by the device 1, the flow advances to step S5, and a game start signal is supplied to the game performing device 23 so as to start the game. A game form is selected by the device 1. However, when the item "PERFORM GAME" is selected immediately after the game has been drawn up, since the corresponding game is set to be executed, the game form need not be designated. Assuming that the battle form game (game I) is selected, a game I selection instruction is supplied from the controller 2 to the selector 5, and the game algorithm memory 6 is connected to the device 23 through the selector 5. A description will be made with reference to the battle form game. Thus, the game algorithm stored in the memory 6 is executed by using various parameters set in the memory group 14.

Figure 20 shows the overall flow chart of the game performing operation. In step S51, previous processing is performed. In the previous processing, i.e., initialization processing, a game name is displayed, a start sound is produced, an initial screen is displayed, and the number of stages is set. Then, the apparatus is changed to a HALT state. That is, since an interrupt signal is generated from the controller 15 at 16-msec intervals, the apparatus stands by until the signal is received. The controller 15 thus performs one cycle of processing every 16 msec. In one cycle of processing, as can be seen from Figure 20, every time processing (steps S52, S54, S56 and S58) is executed for every interruption, character movement processing (steps S53 and S57) is executed every two interruptions, character dump check processing (steps S55 and S59) is executed every other two interruptions, and scrolling processing (step S60) is executed once every four interruptions. The flow shown in Figure 20 is executed for every stage.

Figure 21 shows the every time processing in detail. In step S61, a dump flag generated from the controller 15 is stored in the area A10 of the memory group 14 in accordance with the flow chart of Figure 40. That is, hardware comprises a function for automatically generating the dump flag when the character patterns overlap each other, and the flag is stored in this step. Therefore, when the dump flag is "1", some characters are dumped. In step S62, a timer count operation shown in the flow chart of Figure 41 is performed. Timer data set (although timer setting operation is not described, an initial value thereof is stored in an area B1 of the memory group 14, and a current value is stored in an area B2) for each stage is decremented (if the timer is set). In addition, a disappearance limit counter (area C5) and a movement timer (area C15) determined by the movement speed are updated. Subsequently, in step S63, key input is permitted. Therefore, the key input for playing the game is enabled in this step. Upon depression of a break key, the game is interrupted, and the initial menu screen is displayed. In step S64, counters are checked, as shown in the flow chart of Figure 42. First, it is checked if the number of self characters stored in the area A6 of the memory group 14 becomes "0".If the number of self characters has reached "0", game over processing is performed. In step S64, it is checked if the total number of rival characters has become "0". If the total number of rival characters reaches "0", the current stage ends, and the game advances to the next stage. Furthermore, in step S64, a current score stored in the area A8 is compared with a previous highest score stored in an area A9, and if the current score is larger than the previous highest score, the current score is replaced with the previous highest score in step S64. In step S65, an effect sound generation instruction is produced, and processing for changing parameters by key input received in step S63 is performed.

The character movement step will be described in more detail with reference to Figure 22. Although only one routine from START to END is shown in Figure 22, the same processing is repeated for all the preset characters in practice. It is checked in step S71 if the character is being displayed. This can be determined by checking if the y coordinate data stored in the area 21b of the table 21 is located within the display area on the display screen. Alternatively, it can be determined by accessing a status flag of the memory group 14. If the character is being displayed, the flow advances to step S72, and otherwise, the flow advances to step S80. It is checked in step S72 by accessing the area C15 of the memory group 14 if a movement limit for the character has passed. If NO in step S72, the flow ends. However, if YES in step S72, the movement timer in the area C15 is reset in step S73, and the flow advances to step S74. In step S74, coordinates of a new position are calculated. The new coordinates can be calculated from the area C20 of the memory group 14 shown in Figure 18. It is checked in step S75 if the next coordinates fall within the preset display area defined by the coordinate data XL, XR, YU and YD stored in the area C8. If NO in step S75, the flow advances to step S76, and out-of-area processing for causing the character to disappear or displaying the character at the opposite side of the screen or the like is performed in accordance with the out-of-area processing data stored in the area C16. However, if YES in step S75, the character pattern is selected in accordance with the movement direction of the character in step S77. In other words, the character is directed, as shown in Figure 23, in accordance with an increase or a decrease in x and y coordinates of the next display position stored in the area C20 of the table shown in Figure 18 as compared to the current display position. For example, when the character is moved from the start point (2,4) to a position designated by the area C20b, since both the x and y coordinates are increased, i.e., (+,+), the character is directed in an upward right direction. This direction corresponds to a character pattern direction of one of eight directions shown in Figure 8, and in the case of the upward right direction, the character directing the

upward right direction is selected. That is, the character is detected from the table 20 in accordance with the status flag C23 of the character, and the corresponding character pattern is read out by accessing the table 21 and the generator 22. In step S78, the character displayed at the current coordinates is deleted, and a new character is displayed at new coordinates, in step S79. The status flag corresponding to the new display character is set in the status flag C23. However, the character is not immediately shifted to the next preset junction shown in Figure 18, but is gradually moved to the junction through steps set in the area C21.

However, if NO in step S71, it is checked in step S80 if an appearance time is reached. That is, it is checked if an appearance timer in the area C9 has reached "0". If YES in step S80, an appearance position of the character is set by referring the content of the area C9 in step S81. It is checked in step S82 if the coordinates of the set appearance position fall in the preset display area. If NO in step S82, the character is not displayed, and if YES in step S82, it is displayed. In step S84, an initial value of the area C14 is set in an area C15 of the movement timer, thus setting the movement timer.

The dump check flow (steps S55 and S59) will be described in more detail with reference to Figure 24. Although only a single flow chart from START to END is shown in Figure 24, the same processing is repeated for all the combinations of the present characters in practice. In the flow chart of Figure 24, a description will be made with respect to certain two characters. It is checked in step S91 if the dump flag is set in the area A10. If NO in step S91, since no characters overlap, the dump check flow shown in Figure 24 is not executed. However, if YES in step S91, the flow advances to step S92, and it is checked if the corresponding characters are displayed. If NO in step S92, i.e., if one of the characters is not displayed, no processing is performed for this combination of characters. If YES in step S92, i.e., if both the characters are displayed, the flow advances to step S93, and it is checked if these characters overlap with each other. If NO in step S93, the flow ends. However, if YES in step S93, the flow advances to step S94, and it is judged in accordance with the area C13 which character is stronger, and the weaker character is deleted. For example, if the arms of a self character stronger than a rival character overlap its arms, the rival character is deleted. Note that when the characters have the same strength, both are deleted. In step S95, a defeat character of the deleted character is read out and displayed. The defeat character is read out by referring the area C2 so as to sequentially designate the tables 20 and 21 and the generator 22. In step S97, a defeat sound name is designated by referring to the area C7, and is set in the generator 25. In step S98, a score of the defeated character set in the area C3 is added to a self score "SCORE". In step S99, the number of appearance of the deleted character stored in the area C16 is decremented by one. Thus, the dump processing of the two characters ends, and another processing for another two characters is then performed. For example, if four characters No. 1, No. 2, No. 3 and No. 4 are provided, the dump processing must be performed six times in combinations of No. 1/No. 2, No. 1/No. 3, No. 1/No. 4, No. 2/No. 3, No. 2/No. 4 and No. 3/No. 4.

Finally, in the scrolling processing of step S60 of Figure 20, it is checked if scrolling must be performed. If scrolling must be performed, processing for scrolling the screen in a designated direction is performed. A detailed description thereof will be omitted because this technique is known to those skilled in the art.

When the game is over, the device 23 supplies a game performing end signal to the controller 2.

The game performing operation has been described. Although not the total of game operations, the main features of the present invention can be understood from the above description.

It should be noted that in the above embodiment, separate processors are used as the controller 2, the processors 3, 10, 11, 12 and 13 and the device 23. However, a single processor can serve for the above processors.

Effects of the present invention will be described below:

a. For example, since a desired game can be drawn up simply by inputting parameters needed for a game to be drawn up in accordance with predetermined procedures represented by menu screens, various games suited to the capability and preference of a user can be drawn up and his creative desires can be satisfied at the same time.

b. When a character is drawn up, since a pattern can be shifted within a character block, the pattern can be easily corrected.

c. Since a function for returning to previous menu screens is provided, preset data can be easily corrected. Furthermore, since the menu screens can be returned in units of blocks, a desired menu screen can be quickly displayed, resulting in easy game draw up processing.

d. In order to set a movement course of a character, if a start point and junctions are determined upon viewing the screen, the character can be displayed and moved along the determined course at a predetermined speed, thus easily setting the desired course of the user.

e. Since a character pattern movement direction can be automatically selected and displayed in accordance with the movement direction of the character, a cumbersome operation for setting different character patterns for each movement direction can be avoided.

CLAIMS

1. An electronic game apparatus using an animation display, comprising:

game form setting means for designating basic game algorithm data of a desired game of a predetermined form which has been prepared in advance;

background image setting means for setting background image data to be added to the designated basic game algorithm data;

character setting means for setting character data to be added to the designated basic game algorithm data;

game story setting means for setting game story data to be added to the designated basic game algorithm data; and

game performing means for adding the background image data, the character data and the game story data to the designated basic game algorithm data so as to be displayed on the animation display, thus performing the game in accordance with the predetermined form.

2. An apparatus according to claim 1, wherein said story setting means sets game story data for a plurality of game stages.

3. An apparatus according to claim 1, wherein said apparatus further comprises:

effect sound setting means for setting effect sound data to be added to the designated basic game algorithm data.

4. An apparatus according to claim 1, wherein said game story setting means prepares story data of a predetermined style in advance.

5. An apparatus according to claim 3, wherein said effect sound setting means prepares predetermined effect sound data in advance.

6. An apparatus according to claim 1, wherein said game story setting means includes means for setting movement of a character defined by the preset character data on the display.

7. An apparatus according to claim 1, wherein said character setting means includes means for setting a plurality of different character data.

8. An apparatus according to claim 7, wherein said game performing means includes means for performing different operations with respect to the plurality of different character data.

9. An apparatus according to claim 1, wherein said character setting means includes means for displaying the character data within a block of N × M dots, means for setting a desired character pattern by turning on/off the dots in the block, and means for shifting the character pattern in the block.

10. An apparatus according to claim 1, wherein said apparatus further comprises:

menu display means for displaying a plurality of menus related to setting of data in a predetermined order on the display.

11. An apparatus according to claim 10, wherein said apparatus further comprises:

menu display back means for reversing a menu display order by said menu display means.

12. An apparatus according to claim 11, wherein said menu display means comprises a plurality of blocks each having a predetermined number of menus, and said menu display back means reverses the menu display order by said menu display means in units of blocks.

13. An apparatus according to claim 6, wherein said game story setting means includes means for setting a movement course along which the character defined by the character data is moved on the display.

14. An apparatus according to claim 13, wherein said game story setting means includes means for repeatedly monitoring that the character defined by the character data is moved on the display in accordance with the preset movement course.

15. An apparatus according to claim 6, wherein said game story setting means includes means for setting a movement course and a movement speed of the character defined by the character data on the display.

16. An apparatus according to claim 15, wherein said game story setting means includes means for repeatedly monitoring that the character defined by the character data is moved on the display in accordance with the preset movement course and speed.

17. An apparatus according to claim 13, wherein said game story setting means includes means for displaying that the character defined by the character data is moved on the display along the preset movement course from a start position when the start position of the movement course is designated.

18. An electronic game apparatus using an animation display, comprising:

game form setting means for designating basic game algorithm data of a desired game of a predetermined form which has been prepared in advance;

background image setting means for setting background image data to be added to the designated basic game algorithm data;

character setting means for setting standard-pattern character data and modified-pattern character data obtained by providing a directivity to the standard-pattern character data with respect to a single character as character data to be added to the preset basic game algorithm data;

game story setting means for setting game story data to be added to the designated basic game algorithm data; and

game performing means for adding the background image data, the character data and the game story data to the designated basic game algorithm data so as to be displayed on the animation display, thus performing the game in accordance with the predetermined form.

19. An apparatus according to claim 18, wherein said game story setting means includes means for setting a movement of a character defined by the preset standard-pattern character data on the display.

20. An apparatus according to claim 19, wherein said game story setting means includes means for automatically selecting the modified-pattern character data having the directivity corresponding to a movement direction thereof when the character defined by the preset standard-pattern character data is moved in an arbitrary direction on the display, so that the character defined by the modified-pattern character data is displayed to be moved on the display.

21. An electronic game apparatus, substantially as hereinbefore described with reference to the accompanying drawings.

Printed in the UK for HMSO, D8818935, 1 86, 7102.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.