(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 802 508 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
04.01.2006 Bulletin 2006/01

(21) Application number: **96926619.6**

(22) Date of filing: **09.08.1996**

(51) Int Cl.:
A63F 13/00 *(2006.01)* ***G06F 3/033*** *(2006.01)*

(86) International application number:
PCT/JP1996/002267

(87) International publication number:
WO 1997/006510 (20.02.1997 Gazette 1997/09)

(54) **INPUT DEVICE FOR IMAGE FORMATION METHOD AND ITS APPARATUS**

EINGABEGERÄT FÜR BILDERZEUGUNGSVERFAHREN UND -GERÄT

DISPOSITIF D'ENTREE POUR PROCEDE DE FORMATION D'UNE IMAGE ET APPAREIL CORRESPONDANT

(84) Designated Contracting States:
DE ES FR GB IT

(30) Priority: **10.08.1995 JP 20484895**

(43) Date of publication of application:
22.10.1997 Bulletin 1997/43

(73) Proprietor: **SEGA ENTERPRISES, LTD.**
Tokyo 144 (JP)

(72) Inventors:
- **WATARI, Juro**
Ohta-ku,
Tokyo 144 (JP)
- **SONODA, Yoshihiro**
Ohta-ku,
Tokyo 144 (JP)

(74) Representative: **Style, Kelda Camilla Karen et al**
Page White & Farrer,
54 Doughty Street
London WC1N 2LS (GB)

(56) References cited:
WO-A-94/24685 **JP-A- 6 277 363**

EP 0 802 508 B1

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

[0001] The present invention relates to an image generation technique for use in game units, simulators, and the like, and particularly to a technique for generating images (hereinafter termed "virtual images") obtained when an object present in a virtual three-dimensional space (hereinafter termed "virtual space") is projected (by perspective projection) onto a two-dimensional plane which corresponds to a prescribed visual point.

BACKGROUND ART

[0002] In recent years, game units and simulators equipped with on-board virtual image generation apparatuss which make it possible for movable objects (objects) which move through three dimensional space to combat each other have been developed. Such virtual image generation apparatuss are usually equipped with a virtual image generation apparatus main unit that houses a computer unit for executing stored programs, an input device for sending control signals to the computer unit to instruct it to move objects displayed on the screen within the virtual image, a display for displaying the virtual images generated by the computer unit according to the program sequence, and a sound device for generating sounds according to the program sequence.

[0003] Examples of devices with the architecture described above include driving game unit with a car race theme, in which cars compete with enemy cars on a circuit, and simulators which re-create the experience of piloting a helicopter or airplane. In this type of device, highly realistic simulation of car or helicopter movement is extremely important. For example, in a driving game like that depicted in Fig. 8A, input devices which resemble an actual car steering wheel, accelerator pedal, and brake pedal are used. In a helicopter or other simulator input device are processed by the CPU (central processing unit) of the computer unit. The computer unit repeatedly performs calculations to assign relative position within the virtual space to the objects, including data for enemy object motion when enemy objects are also present.

[0004] As players become more skilled at playing games, it has become necessary to go beyond conventional motion and develop movable objects such as player-controlled robots, humans, and the like. Particularly in the field of game devices, games are being developed in which objects not only move in two dimensions over a terrain created in virtual space (hereinafter termed "virtual terrain"), but also jump up from a virtual terrain in order to jump over another character or engage in fights in midair.

[0005] However, input devices for conventional virtual image generation apparatuss, while suitable for controlling two-dimensional motion of objects through a virtual space, are not adapted to controlling three-dimensional motion such as jumping. For example, in the driving games mentioned earlier, the steering wheel (which is the principal means of control) controls the movable object in the sideways direction (as viewed from the player's visual point), while the accelerator and brake pedals control motion in the forward direction; there is no way to control motion of the movable object in the vertical direction. Similarly, in flight simulators, a single control lever is used to control all motion of the movable object in three directions, the forward direction, sideways direction, and vertical direction.

[0006] In combat-style game units, the game unit must afford control sufficient to permit agile motion in order to avoid an enemy attack. In such cases, a special control button or control lever to control jumping can be provided, but this makes operation complicated and does not allow the action to be transmitted to the game unit with the same sense of speed that the player desires. In addition, excessively feature-laden input devices entail higher costs.

[0007] By way of improving control, a video game unit which affords simple game control using two control levers is taught in Japanese Laid-Open Patent Application 6-277363. In this example of the prior art, thrust vectors are assigned according to the control lever inclination of each of the two control levers, and the two vectors are synthesized to produce complex actions.

[0008] However, in this example of the prior art, it is difficult to move the object rapidly in the desired direction through synthesis of the two vectors, and it is not possible to move the object freely to a desired position in three-dimensional space.

[0009] In order to solve this problem, it is an object of the present invention to provide a virtual image generation method and device which allow a movable object to be moved freely and without input error in three dimensions within a virtual space, and to a device therefore.

Japanese Patent No. JP-6-277363 describes a video game system for playing a game while controlling a moveable game object. A player operates first and second steering unit with his or her right and lefts hands to input first and second proportion vectors for first and second proportion units in the moveable game object. A steering information computing unit is responsive to the first and second proportion vectors to compute direction and rotation information indicating the direction and manner in which the moveable game object moves. The direction and rotation information are output at a steering signal to move the moveable game object within the game space.

The present invention provides an image generation method according to claim 1, and an image generation apparatus according to claim 3.

Input devices for generating codes associated with the operating direction thereof include, for example, control levers and joysticks. The decoding means may, for example, be a CPU or controller, and the image generation means may, for example, be a CPU, geometalyzer, or graphic controller.

[0010] In one embodiment, in the event that a combination of codes input from the input devices matches a prescribed combination (for example, when the left input means is moved forward and the right input means is lowered towards the player), the assigned movement is such that the movable object rotates while remaining in the same position within the virtual space.

[0011] In one embodiment, in the event that a combination of codes input from the input devices matches a prescribed combination (for example, when the left input means is moved forward diagonally towards the right and the right input means is lowered towards the right), the assigned movement is such that the movable object moves in the lateral direction along a prescribed circle which is centered upon a prescribed central axis within the virtual space.

[0012] In one embodiment, the input devices are control levers which generate a center position code when moved to a prescribed location and which generate a different code when moved in any of eight directions from the prescribed location.

[0013] In another embodiment, switch or control buttons that can detect a neutral position and eight directions are substituted for the control levers.

[0014] In accordance with one embodiment, numerous code combinations are afforded by the control attitudes of the plurality of control means. By associating these various combinations with various motions of movable objects in virtual space, the movable objects can be induced to undergo complex motion. Therefore, the movement of a movable object can be clearly defined through selection of a given control attitude, and by perspective projection of the movable object, the virtual terrain, and the like with reference to this defined movement, virtual images adapted to game units, simulators, and the like can be generated.

[0015] Even under conditions where there is a high likelihood of unintended operation, such as when the player operates the input device in an intuitive fashion in order to dodge a bullet, assigning movable object movements in such a way that such movements approximate those movements which are presumably intended by the player reduces the likelihood of unintended operation, thereby reducing the demands placed upon the player. By assigning three dimensional movements, such as jumping by a movable object, in addition to two-dimensional movements, it becomes possible to move the movable object in three dimensions.

[0016] In one embodiment, rotation in a fixed position is assigned to a specific combination of operations, thereby allowing the orientation of the movable object to be changed without changing its two dimensional position in a virtual space.

[0017] In one embodiment, orbit around a prescribed central axis is assigned to a specific combination of operations, making possible actions such as orbiting around an enemy character.

[0018] In one embodiment, control levers are employed as the control means, and each control lever can maintain nine different control attitudes. Thus, the use of a plurality of control levers affords a number of combinations sufficient to allow control of complex movements by a movable object.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019]

Fig. 1 is a simplified block diagram of a game unit pertaining to an embodiment of the present invention;

Fig. 2 is an illustrative diagram of input devices (control levers) pertaining to the embodiment;

Fig. 3 is an illustrative diagram of the control lever operation method;

Fig. 4 is an assignment diagram showing the control attitudes of the left and right control levers and the associated movements of a movable object in the embodiment;

Fig. 5 is a flow chart illustrating the operation of the game unit pertaining to the embodiment;

Fig. 6 is a diagram which illustrates embodiment 1;

Fig. 7 is a diagram which illustrates embodiment 2; and

Fig. 8 is a diagram depicting an input device for a conventional virtual image generation apparatus.

BEST MODE FOR CARRYING OUT THE INVENTION

[0020] Favorable embodiments of the present invention will be described below with reference to the drawings.

(1) Description of Structure

[0021] A structural diagram of a game unit which represents one embodiment of the present invention is presented in Fig. 1. In this embodiment, a robot serves as the movable object. Controlled by the player, the robot moves freely within the virtual space, engaging in combat with enemy robots.

[0022] As shown in Fig. 1, the game unit 1000 comprises the following basic structural elements: a game unit main body 10, an input device 11, an output device 12, a TV monitor 13, and a speaker 14.

[0023] The input device 11 is provided with control levers which are operated with the player's left and right hands in order to control the movement of the robot. The output device 12 is provided with various types of lamps which notify the player of the operational status of the unit. The TV monitor 13 displays the combat game image; a head mounted display (HMD), projector, or the like may be used in place of a TV monitor.

[0024] As image generation means, the game unit main body 10 has a counter 100 and a CPU (central processing unit) 101; it is also equipped with ROM 102, RAM 103, a sound device 104, an I/O interface 106, a scroll data processor 107, a coprocessor 108, terrain data ROM 109, a geometalyzer 110, shape data ROM 111, a displaying device 112, texture data ROM 113, texture map RAM 114, a frame buffer 115, an image synthesis device 116, and a D/A converter 117. The game unit main body 10 generates new virtual images at prescribed intervals (for example, each 1/60th of a second, corresponding to the vertical sync cycle of the television format).

[0025] The CPU 101, which serves as the decoding means, is connected via buslines to the counter 100, which stores initial values, to the ROM 102, which stores the program for the game sequence and image generation, to the RAM 103, which stores temporary data, and to the sound card 104, I/O interface 106, scroll data processor 107, coprocessor 108, and geometalyzer 110.

[0026] The RAM 103 temporarily stores data required for polygon data coordinate conversion and other functions, and stores various commands for the geometalyzer (such as object display commands), the results of matrix operations during conversion process operations, and other data.

[0027] When the player enters control signals through the input device 11, the I/O interface 106 issues interrupt requests to the CPU 101; when the CPU 101 sends data for lamp display, this data is sent to the output device 12.

[0028] The sound card 104 is connected to a speaker 14 through a power amplifier 105. Audio signals output by the sound card 104 are amplified by the power amplifier 105 and delivered to the speaker 14.

[0029] The ROM 111 stores the polygon data required to generate virtual images of various physical objects such as the player's robot, enemy robots, bomb explosion images, and elements of the virtual terrain such as obstacles, background, and topographical features.

[0030] The ROM 109 stores shape data for physical objects (buildings, obstacles, topographical features, and the like) concerning which it is necessary to make overlap decisions, i.e., whether an object should collide with another topographical feature, or be hidden by a topographical feature. In contrast to the relatively detailed polygon data groupings for image display stored in the ROM 111, the data groupings stored in the ROM 109 comprise rough units sufficient to perform overlap decisions and the like. For example, topographical feature data might include an ID for each surface which defines a topographical feature, and what is termed relationship of this ID and topographical feature surface is put in table form and stored in the ROM 111.

[0031] What is termed polygon data are data groupings which are sets comprising a plurality of apices, and which indicate the apices of polygons (usually triangles or quadrangles), the elements that make up the shape of a physical object, as relative coordinates or absolute coordinates.

[0032] In order to generate virtual images, a coordinate system (world coordinate system) which indicates the relative positions of objects, obstacles, and other physical objects in a virtual space, must be converted to a two-dimensional coordinate system (visual point coordinate system) which represents the virtual space viewed from a designated visual point (for example, a camera or the like). The visual point is set at some prescribed position (for example, diagonally above the object) from which the object to be controlled is visible. Thus, the visual point coordinates change in accordance with the object coordinates. The object coordinates are sent as control signals from the input device 11 to the CPU 101.

[0033] The entire input device 11 is depicted in Fig. 2A. As may be discerned from the drawing, the input device 11 comprises a left control lever 11L operated with the player's left hand and a right control lever 11R operated with the right hand. Each control lever has a total of nine control attitudes, forward and back, left and right, diagonal, and neutral (see Fig. 3). Control signals which correspond to the various control attitudes are output as digital signal codes. As shown in Fig. 2B, the control levers are equipped with shot triggers 11S and turbo triggers 11T for acceleration; codes are output when these are depressed.

[0034] Fig. 4 shows the object movement assignments for the control attitudes. Since each control lever has nine codes, simultaneous operation of both the left and right control levers gives a total of 81 possible combinations (= 9 possibilities x 9 possibilities). If a direction of motion of an object which is to be moved at the next interval is assigned to each combination, a total of 81 actions can be specified using two control levers.

[0035] Assignments should be made in such a way that the direction of motion of the actual object reflects as closely as possible the direction in which the player intuitively tries to move the movable object. However, since horizontal movement on the true horizontal, movement in a diagonal direction, and rotating, as well as jumping in the perpendicular direction from a horizontal plane in virtual space (the z-direction in the world coordinate system) are performed, these special actions should be assigned to prescribed control attitudes.

[0036] When the player is attacked by enemies, he or she moves the control levers to instinctively dodge enemy bullets in order to avoid the attack. With the control lever assignments in this embodiment, movement assignments are made in such a way that the movements intended by the player are reflected in the movements of the object, even for those actions which the player performs reflexively.

[0037] Once one of the code combinations indicated in Fig. 4 has been input from the input device 11, the CPU 101, following the program assigned in the manner indicated in Fig. 4, generates the visual point coordinates and object coordinates for the next interval. Once these coordinates have been set, the CPU 101 makes collision determinations and overlap determinations for the physical objects.

[0038] Objects, obstacles, and other physical objects are composed of a plurality of polygon data. For each physical object, a certain apex of a polygon which is an element of the physical object is selected as the origin, the entire shape is decided using a coordinate system which indicates the coordinates of the other apices (body coordinate system), and the data for polygons which make up the physical object is associated. In order to enable display of an explosion image when an object or obstacle is hit by a bullet or light ray, it is necessary to compute the relative positions of the physical objects and determine whether the physical objects have collided. To obtain relative position for a physical object represented by body coordinates, conversion to the prescribed coordinate system which makes up the virtual space (world coordinate system) must be made. Once the relative position for each physical object has been determined, it becomes possible to determine whether the physical objects collide with each other.

[0039] In order to enable show-through display of an obstacle when, from the visual point from which a virtual space is observed, an object or the like passes behind the obstacle, it is necessary to perform a determination of overlap status for the physical objects. To do this, the physical objects in the virtual space are converted to the coordinate system for viewing from the visual point, and a relative vector for the obstacle and object and a line-of-sight vector for the object and visual point are calculated. Once the angles of the two vectors have been computed, it can be determined whether the object should be hidden by obstacle or not. Since these computations entail coordinate conversion, matrix operations which include floating decimal point operations are required. Matrix operations are performed by the coprocessor 108 making reference to terrain data and the like stored in ROM 109; as a result of the operations, the CPU 101 makes a collision determination or overlap determination.

[0040] A further requirement for image display is that physical objects in a virtual space be projected onto a two-dimensional plane which constitutes the field of vision in a manner similar to physical objects present in a virtual space observed from a given visual point (for example, camera photography). This is termed perspective projection, and the coordinate conversion performed through matrix operations for perspective projection is termed perspective conversion. It is the geometalyzer 110 that executes perspective conversion to produce the virtual image which is actually displayed.

[0041] The geometalyzer 110 is connected to the shape data ROM 111 and the displaying device 112. The geometalyzer 110 is provided by the CPU 101 with data indicating the data required for perspective conversion as well as with the matrix data required for perspective conversion. On the basis of the matrix provided by the CPU 101, the geometalyzer 110 performs perspective conversion on polygon data stored in the shape data ROM 111 to produce data converted from the three-dimensional coordinate system in virtual space to the field-of-vision coordinate system. At this time, if it is necessary to display an explosion image as a result of a collision determination by the CPU 101, polygon data for the explosion image is used.

[0042] The displaying device 112 applies texture to the converted field-of-vision coordinate system shape data and outputs the result to the frame buffer 115. If, as a result of an overlap determination, the object or the like is to be hidden behind an obstacle, a prescribed show-through display (mesh treatment or translucent treatment) is performed. To apply textures, the displaying device 112 is connected to the texture data ROM 113 and the texture map RAM 114, and is also connected to the frame buffer 115.

[0043] The scroll data processor 107 computes text and other scroll screen data (stored in ROM 102). The image synthesis device 116 imposes text data output from the processor 107 onto the image data provided by the aforementioned frame buffer 115 and re-synthesizes the image. The re-synthesized image data is output to the TV monitor 13 through the D/A converter 117.

(II) Description of Operation

[0044] Next, the operation in this embodiment will be described referring to the flow chart in Fig. 5.

[0045] When the player moves the left control lever 11L, the right control lever, or both, thereby inputting a new control signal to the I/O interface 106, the I/O interface 106 makes an interrupt request to the CPU 101. If there is no interrupt (step S1: NO), the CPU performs other processes (step S2), but if an interrupt request has been made, (step S1: YES), the control signal is acquired.

[0046] In this embodiment, in order to determine a way of doing unintended input or not, the control signal is ascertained for each interval after an interrupt request, and if the same input signal is input eight consecutive times, a determination of correct input is made. To do this, the counter is first set to the initial value n (step S3) and the left control lever control signal and right control lever control signal are input (step S4, S5).

[0047] The CPU 101 compares the value of the control signal input during the previous interval with the currently input control signal value (step S6). If the two do not match (step S6: NO), a determination of unintended input is made, and the CPU awaits the next interrupt request (step 1). If the previous control signal value and the currently input control signal value are equal (step S6: YES), the CPU determines whether the same determination has been made eight times (step S7). If less than eight times (step S7: NO), the counter n is incremented (step S8) and the same procedure is repeated (steps S4 - S7). If the same value has been input eight times (step S7: YES), the system proceeds to generate a virtual image on the basis of the correct control signal.

[0048] In step S9, on the basis of the coordinates of the destination point of the player's robot (object), the CPU 101 creates a perspective conversion matrix, a matrix for perspective conversion of shape data in virtual space into the visual point coordinate system, and provides this to the geometalyzer 110. At the same time, the CPU 101 provides the coprocessor 108 with the terrain data stored in ROM 109 and instructs the coprocessor to perform coordinate conversion for making a collision determination; if a "collision" result is produced, data indicating the necessary polygons is output to the geometalyzer 110. Where vector operations for making an overlap determination have produced an overlap result, the CPU 101 instructs the geometalyzer 110 to produce show-through display.

[0049] In step S10, processing similar to that described in step S9 is performed for the enemy robot. The enemy robot can be made to move in accordance with the program stored in ROM 102, or made to move by another input device controlled by another player.

[0050] In step S11, the data required to designate polygons required for perspective conversion is provided to the geometalyzer 110.

[0051] In step S12, the geometalyzer 110 uses the provided perspective conversion matrix to perform perspective conversion for the designated shape data and supplies the result to the displaying device 112. The displaying device 112 performs texture application and the like for the perspective-converted polygons and outputs the result to the frame buffer 115.

[0052] With the embodiment described above, the control levers output control signals which take the form of digital data, thereby minimizing the likelihood of unintended input. Since movable object movement assignments are made in such a way that objects can be moved correctly, control is facilitated, even in scenes where it is easy to make unintended moves. Special assignments are made for control attitudes which the player is thought unlikely to actually use, such as jumping, rotating, circling an enemy, rapid acceleration, and rapid stop, thereby allowing objects to be moved freely in three dimensions within a virtual space.

(III) Other Embodiments

[0053] The present invention is not limited to the embodiment described above and may be adapted in various ways.

[0054] For example, the input device was equipped with two control levers in the foregoing embodiment, but the present invention may be adapted to any configuration which outputs digital control signals, such as a joystick or control button that can be depressed in eight directions. The number of control directions is not limited to eight; implementation with more or fewer directions is possible.

[0055] Control attitude assignments are not limited to the assignments indicated in Fig. 4 and permit of various modifications in accordance with the specifications of the game unit, simulator, or other unit equipped with the image generating device which pertains to the present invention.

[0056] As the present invention was designed with the principal aim of facilitating control of movable objects in virtual space, the virtual image generation method may employ various image generation methods pertaining to computer graphics.

Examples

[0057] An example in which the game unit 1000 in the aforementioned embodiment of the invention is actually used will be described.

[0058] Fig. 6 depicts example 1 (scene 1), illustrating moves to evade a bullet fired by a combat partner (enemy).

[0059] A of the same figure depicts positional relationships at the instant the bullet is fired by the enemy; positions are viewed from above. As in the previous embodiment, the player manipulates the control levers in the manner shown in Fig. 4(1) to evade the bullet. If control lever assignments have been made in the manner indicated in Fig. 4, the player's object performs a "slow circle counterclockwise to right". As shown in Fig. 6B, the player's object moves so as to circle the enemy. Scene 1, when actually displayed as a virtual image on a monitor, would appear as shown in C of the same figure. Since the visual point of the virtual image rotates with the movement of the player's object, the enemy's displayed position is virtually unchanged as the movement of the player's object to circle the enemy is displayed. This image display minimizes movement of the player's line of sight, reducing the demands placed on the player and allowing the excitement of real combat to be sustained.

[0060] As shown in D of the same figure, to advance the player's object forward after circling the enemy in order to counterattack, the player should lower both control levers in the forward direction (Fig. 4(2)). To perform both rapid circling and forward advance within a short time, the player can on occasion lower the control levers in the manner depicted in Fig. 4(3).

[0061] However, with the assignments indicated in Fig. 4, such frequently used control attitudes are reflected in the object as movements intended by the player, so the player's object can be advanced toward the enemy. That is, when objects are controlled through a synthesis vector for two levers, as described in the Background Art, the operation indicated in Fig. 4(3) can easily give rise to an unintended move; in this example, however, the operation indicated in Fig. 4(3) has the assignment "advance diagonally slightly to left", thereby allowing the object to be moved in the intended direction without slowing the pace of the game sequence.

[0062] Fig. 7 depicts example 2 (scene 2), illustrating an enemy and the player's object circling each other around an obstacle.

[0063] A of the same figure shows the positional relationship of the enemy and the player's object. If only two dimensional movements within the virtual space can be specified, as with conventional game units, only movement around the same obstacle around which the two are trying to circle is possible, thereby slowing the pace.

[0064] With this example, the player's object can be made to "jump" as it circles, as shown in Fig. 4(4), thereby allowing the player's object to jump down on the enemy from above and attack the enemy, as shown in the actual virtual image in Fig. 7B. By making a "forward advance" control move after making the "jump" control move, the player's object can be moved toward the enemy while maintaining the same altitude. This affords a fast-paced game sequence without complicated control moves.

[0065] In the above-described embodiments, a pair of input device control attitudes are combined, and a specific movement is specified for each pair of control attitudes. This reduces the likelihood of unintended moves and facilitates control, thereby making it possible to freely control movable objects within virtual space.

[0066] For selected pairs of control attitudes, three-dimensional movement is facilitated by assigning a jump move, rotation of a movable object may be facilitated by assigning a rotating move, and motion on a circle around a prescribed axis may assigned to facilitate circling around an enemy.

Claims

1. An image generation method for generating images of the movement of a movable object in a virtual three-dimensional space, using a pair of input devices (11) for respective operation by the left and right hands of an operator and each having forward, back, left, right and diagonal control attitudes selectable by the operator, comprising the steps of:

generating, by each of the pair of input devices, a respective code corresponding to the respective selected control attitude;
assigning to the movable object on the basis of the combination of generated codes one of a plurality of movements including movements in a horizontal plane;
generating an image reflecting the assigned movement of the moveable object in the virtual three-dimensional space;
characterised in that: the assigning step includes assigning to the moveable object a jumping movement in a direction perpendicular to said horizontal plane when the combination of generated codes matches a prescribed combination.

2. A machine readable storage medium storing a program which embodies an image generation method-as claimed in claim 1.

3. An image generation apparatus for generating images of the movement of a movable object in a virtual three-dimensional space, comprising:

a pair of input devices (11) for respective operation by the left and right hands of an operator; each input device having forward, back, right, left and diagonal control attitudes selectable by the user; and each input device adapted to generate a code corresponding to the selected control attitude;
decoding means (101) adapted to receive the codes generated by the pair of input devices, and on the basis of the combination of said codes assign to the moveable object one of a plurality of movements including movement in a horizontal plane; and
image generation means (101) adapted to generate an image reflecting the assigned movement of the moveable object in the virtual three-dimensional space;
characterised in that: the decoding means assigns to the moveable object a jumping movement in a direction perpendicular to the horizontal plane when the combination of codes generated by the input devices matches a prescribed combination.

4. An image generation apparatus as defined in claim 3, wherein the image generated is an explosion image.

5. An image generation method as defined in claim 1, wherein the image generated is an explosion image.

Patentansprüche

1. Bilderzeugungsverfahren zum Erzeugen von Bildern der Bewegung eines beweglichen Objekts in einem virtuellen, dreidimensionalen Raum unter Verwendung zweier Eingabevorrichtungen (11) jeweils für die Betätigung durch die linke und rechte Hand einer Bedienperson und die jeweils Vorwärts-, Rückwärts-, Links-, Rechts- und Diagonalsteuerlagen haben, die von der Bedienperson wählbar sind, umfassend die Schritte:

durch jede der beiden Eingabevorrichtungen Erzeugen eines entsprechenden Codes, der der jeweils gewählten Steuerlage entspricht;
auf der Grundlage der Kombination erzeugter Codes Zuweisen einer von mehreren Bewegungen, einschließlich Bewegungen in einer horizontalen Ebene, zum beweglichen Objekt;
Erzeugen eines Bildes, das die zugewiesene Bewegung des beweglichen Objekts in dem virtuellen dreidimensionalen Raum wiedergibt;

dadurch gekennzeichnet, dass der Zuweisungsschritt das Zuweisen einer Sprungbewegung in einer Richtung senkrecht zu der horizontalen Ebene zu dem beweglichen Objekt einschließt, wenn die Kombination erzeugter Codes mit einer vorgeschriebenen Kombination übereinstimmt.

2. Maschinenlesbares Speichermedium, das ein Programm speichert, das ein Bilderzeugungsverfahren verkörpert, wie in Anspruch 1 beansprucht.

3. Bilderzeugungsgerät zum Erzeugen von Bildern der Bewegung eines beweglichen Objekts in einem virtuellen dreidimensionalen Raum, enthaltend:

zwei Eingabevorrichtungen (11) jeweils für den Betrieb durch die linke und rechte Hand einer Bedienperson, wobei jede Eingabevorrichtung Vorwärts-, Rückwärts-, Rechts-, Links- und Diagonalsteuerlagen hat, die durch den Benutzer wählbar sind, und jede Eingabevorrichtung dazu eingerichtet ist, einen der gewählten Steuerlage entsprechenden Code zu erzeugen;
eine Decodiereinrichtung (101), die dazu eingerichtet ist, die durch die beiden Eingabevorrichtungen erzeugten Codes zu empfangen und auf der Grundlage der Kombination dieser Codes dem beweglichen Objekt eine von mehreren Bewegungen, einschließlich der Bewegung in einer horizontalen Ebene, zuzuweisen; und
eine Bilderzeugungseinrichtung (101), die dazu eingerichtet ist, ein Bild zu erzeugen, das die dem beweglichen Objekt zugewiesene Bewegung im virtuellen dreidimensionalen Raum wiedergibt;
dadurch gekennzeichnet, dass die Decodiereinrichtung dem beweglichen Objekt eine Sprungbewegung in eine Richtung senkrecht zu der horizontalen Ebene zuweist, wenn die Kombination der von den Eingabevorrichtungen erzeugten Codes mit einer vorgeschriebenen Kombination übereinstimmt.

4. Bilderzeugungsgerät nach Anspruch 3, bei dem das erzeugte Bild eine Explosionsdarstellung ist.

5. Bilderzeugungsverfahren nach Anspruch 1, bei dem das erzeugte Bild eine Explosionsdarstellung ist.

Revendications

1. Procédé de génération d'image pour générer des images du mouvement d'un objet mobile dans un espace virtuel en trois dimensions, utilisant une paire de dispositifs d'entrée (11) pour une opération respective par les mains gauche et droite d'un opérateur et comportant chacun des attitudes de commande vers l'avant, l'arrière, la gauche, la droite et en diagonale, pouvant être sélectionnées par l'opérateur, comprenant les étapes :

de génération, par chacun de la paire de dispositifs d'entrée, d'un code respectif correspondant à l'attitude de commande sélectionnée respective ;
d'attribution à l'objet mobile, sur la base de la combinaison de codes générée, de l'un d'une pluralité de mouvements incluant des mouvements dans un plan horizontal ;
de génération d'une image reflétant le mouvement attribué de l'objet mobile dans l'espace virtuel en trois dimensions ;
caractérisé en ce que : l'étape d'attribution comporte une attribution à l'objet mobile d'un mouvement de saut dans une direction perpendiculaire audit plan horizontal lorsque la combinaison de codes générée correspond à une combinaison prescrite.

2. Support de stockage lisible en machine stockant un programme qui met en oeuvre un procédé de génération d'image selon la revendication 1.

3. Appareil de génération d'image pour générer des images du mouvement d'un objet mobile dans un espace virtuel en trois dimensions, comprenant :

une paire de dispositifs d'entrée (11) pour une opération respective par les mains gauche et droite d'un opérateur ; chaque dispositif d'entrée ayant des attitudes de commande vers l'avant, l'arrière, la droite, la gauche et en diagonale, pouvant être sélectionnées par l'utilisateur ; et chaque dispositif d'entrée étant adapté à générer un code correspondant à l'attitude de commande sélectionnée ;
un moyen (101) de décodage adapté à recevoir les codes générés par la paire de dispositifs d'entrée, et sur la base de la combinaison desdits codes, à attribuer à l'objet mobile l'un d'une pluralité de mouvements incluant un mouvement dans un plan horizontal ; et
un moyen (101) de génération d'image adapté à générer une image reflétant le mouvement attribué de l'objet mobile dans l'espace virtuel en trois dimensions ;
caractérisé en ce que : le moyen de décodage attribue à l'objet mobile un mouvement de saut dans une direction perpendiculaire au plan horizontal lorsque la combinaison de codes générée par les dispositifs d'entrée correspond à une combinaison prescrite.

4. Appareil de génération d'image selon la revendication 3, dans lequel l’image générée est une image d’explosion.

5. Procédé de génération d’image selon la revendication 1, dans lequel l’image générée est une image d’explosion.

FIG 1

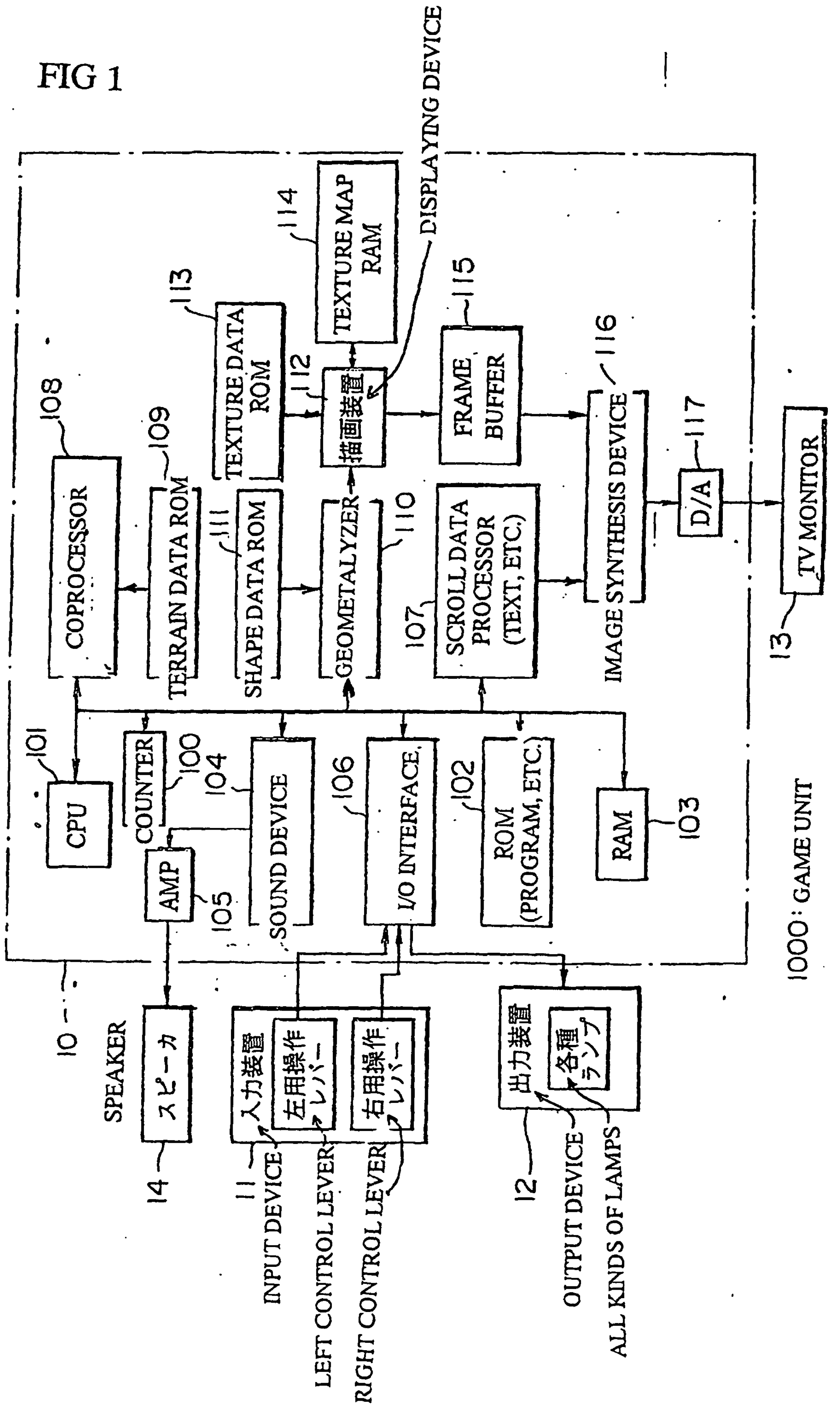
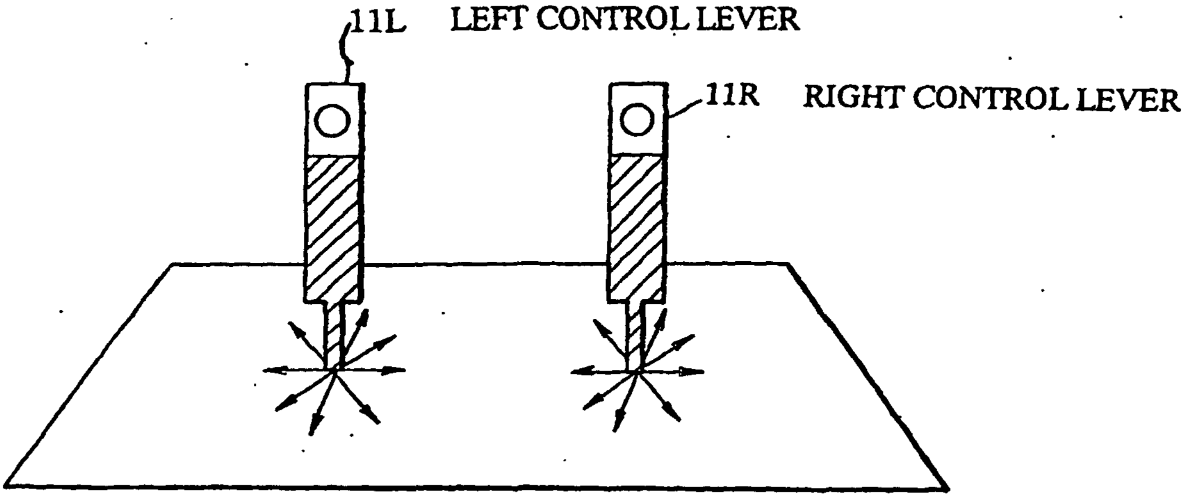

DISPLAYING DEVICE
114
TEXTURE MAP RAM
113
TEXTURE DATA ROM
112
描画装置
115
FRAME BUFFER
116
IMAGE SYNTHESIS DEVICE
117
D/A
TV MONITOR
13
108
COPROCESSOR
109
TERRAIN DATA ROM
111
SHAPE DATA ROM
GEOMETALYZER
110
107
SCROLL DATA PROCESSOR (TEXT, ETC.)
101
CPU
COUNTER
100
104
SOUND DEVICE
106
I/O INTERFACE
102
ROM (PROGRAM, ETC.)
RAM
103
AMP
105
1000: GAME UNIT
10
SPEAKER
スピーカ
14
入力装置
左用操作レバー
右用操作レバー
11
INPUT DEVICE
LEFT CONTROL LEVER
RIGHT CONTROL LEVER
出力装置
各種ランプ
12
OUTPUT DEVICE
ALL KINDS OF LAMPS

FIG.2A A VIEW OF ENTIRE INPUT DEVICE



SIDE VIEW OF INPUT DEVICE

FIG.2B

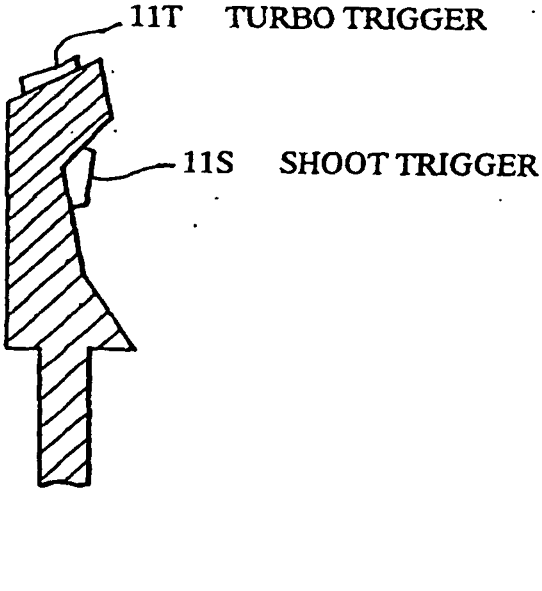


FIG.3

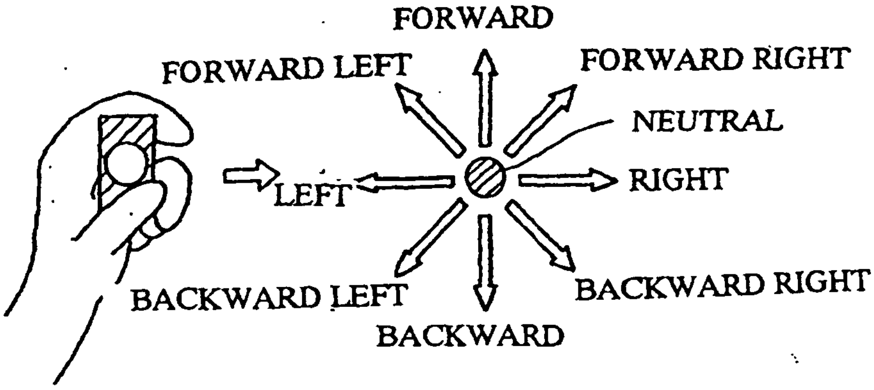

FORWARD
FORWARD LEFT
FORWARD RIGHT
NEUTRAL
LEFT
RIGHT
BACKWARD LEFT
BACKWARD RIGHT
BACKWARD

FIG.4

CONTROL ATTITUDE		MOVEMENT	CONTROL ATTITUDE		MOVEMENT	CONTROL ATTITUDE		MOVEMENT
RIGHT	LEFT		RIGHT	LEFT		RIGHT	LEFT	
↑	•	CIRCLE FORWARD CLOCKWISE	↑	↑	FORWARD ADVANCE (2)	↑	↗	ADVANCE DIAGONALLY SLIGHTLY TO RIGHT
↑	→	CIRCLE FORWARD CLOCKWISE	↑	↘	CIRCLE FORWARD CLOCKWISE	↑	↓	ROTATE CLOCKWISE IN POSITION
↑	↙	CIRCLE FORWARD CLOCKWISE	↑	←	CIRCLE FORWARD CLOCKWISE	↑	↖	ADVANCE DIAGONALLY SLIGHTLY TO LEFT
↗	•	CIRCLE FORWARD CLOCKWISE	↗	↑	BACK UP DIAGONALLY SLIGHTLY TO RIGHT	↗	↗	ADVANCE DIAGONALLY TO RIGHT
↗	→	SLOW CIRCLE CLOCKWISE TO RIGHT	↗	↘	CIRCLE CLOCKWISE TO RIGHT	↗	↓	BACK UP CLOCKWISE
↗	↙	ROTATE CLOCKWISE IN POSITION	↗	←	STOP	↗	↖	FORWARD ADVANCE
→	•	SLOW RIGHT ADVANCE	→	↑	CIRCLE FORWARD COUNTERCLOCKWISE	→	↗	SLOW CIRCLE COUNTERCLOCKWISE TO RIGHT
→	→	RIGHT ADVANCE	→	↘	SLOW CIRCLE CLOCKWISE TO RIGHT	→	↓	BACK UP CLOCKWISE
→	↙	STOP	→	←	STOP	→	↖	STOP
↘	•	SLOW RIGHT ADVANCE	↘	↑	CIRCLE FORWARD COUNTERCLOCKWISE	↘	↗	SLOW CIRCLE COUNTERCLOCKWISE TO RIGHT (1)
↘	→	SLOW CIRCLE COUNTERCLOCKWISE TO RIGHT	↘	↘	BACK UP DIAGONALLY TO RIGHT	↘	↓	BACK UP DIAGONALLY SLIGHTLY TO RIGHT
↘	↙	BACK UP	↘	←	STOP	↘	↖	ROTATE COUNTERCLOCKWISE IN POSITION
↓	•	CIRCLE COUNTERCLOCKWISE BACK	↓	↑	ROTATE COUNTERCLOCKWISE IN POSITION	↓	↗	BACK UP COUNTERCLOCKWISE
↓	→	CIRCLE COUNTERCLOCKWISE BACK	↓	↘	BACK UP DIAGONALLY SLIGHTLY TO LEFT	↓	↓	BACK UP
↓	↙	BACK UP DIAGONALLY SLIGHTLY TO LEFT	↓	←	BACK UP COUNTERCLOCKWISE	↓	↖	BACK UP COUNTERCLOCKWISE
↙	•	SLOW ADVANCE TO LEFT,	↙	↑	ROTATE COUNTERCLOCKWISE IN POSITION	↙	↗	ROTATE COUNTERCLOCKWISE IN POSITION
↙	→	JUMP	↙	↘	BACK UP	↙	↓	BACK UP DIAGONALLY SLIGHTLY TO LEFT
↙	↙	BACK UP DIAGONALLY TO LEFT	↙	←	SLOW CIRCLE COUNTERCLOCKWISE TO LEFT	↙	↖	CIRCLE COUNTERCLOCKWISE TO LEFT
←	•	SLOW ADVANCE TO LEFT	←	↑	CIRCLE FORWARD COUNTERCLOCKWISE	←	↗	JUMP
←	→	JUMP (4)	←	↘	JUMP	←	↓	BACK UP CLOCKWISE
←	↙	SLOW CLOCKWISE CIRCLE TO LEFT	←	←	ADVANCE LEFT	←	↖	SLOW CIRCLE COUNTERCLOCKWISE TO LEFT
↖	•	SLOW ADVANCE TO LEFT	↖	↑	ADVANCE DIAGONALLY SLIGHTLY TO LEFT (3)	↖	↗	FORWARD ADVANCE
↖	→	JUMP	↖	↘	ROTATE CLOCKWISE IN POSITION	↖	↓	RAPID BACKUP CLOCKWISE
↖	↙	CIRCLE CLOCKWISE TO LEFT	↖	←	SLOW CIRCLE CLOCKWISE TO LEFT	↖	↖	ADVANCE DIAGONALLY TO LEFT
•	•	STOP	•	↑	SLOW CIRCLE FORWARD COUNTERCLOCKWISE	•	↗	SLOW CIRCLE FORWARD COUNTERCLOCKWISE
•	→	SLOW ADVANCE TO RIGHT	•	↘	SLOW ADVANCE TO RIGHT	•	↓	BACK UP CLOCKWISE
•	↙	SLOW ADVANCE TO LEFT	•	←	SLOW ADVANCE TO LEF	•	↖	SLO ADVANCE TO LEFT

FIG.5

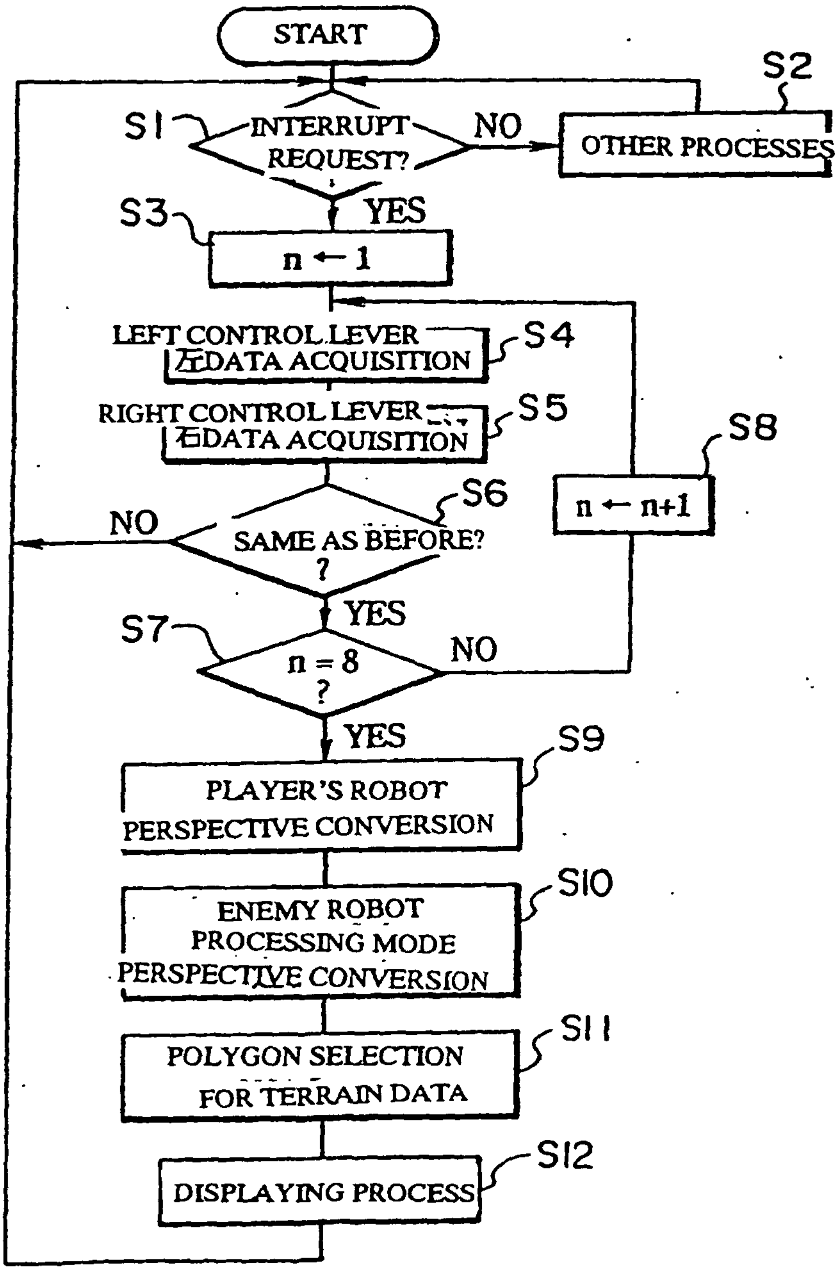

START
S1
INTERRUPT REQUEST?
NO
S2
OTHER PROCESSES
YES
S3
n ← 1
LEFT CONTROL LEVER 左DATA ACQUISITION
S4
RIGHT CONTROL LEVER 右DATA ACQUISITION
S5
S6
NO
SAME AS BEFORE?
YES
S8
n ← n+1
S7
n = 8 ?
NO
YES
S9
PLAYER'S ROBOT PERSPECTIVE CONVERSION
S10
ENEMY ROBOT PROCESSING MODE PERSPECTIVE CONVERSION
S11
POLYGON SELECTION FOR TERRAIN DATA
S12
DISPLAYING PROCESS

FIG.6A SCENE 1: ILLUSTRATION OF PLAYER AND ENEMY POSITIONS (VIEWED FROM ABOVE)

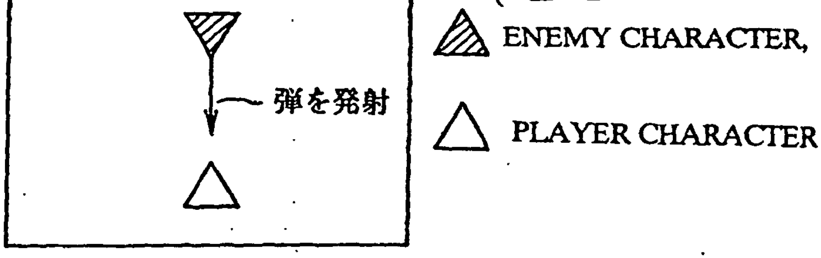


FIG.6B SCENE 1: ILLUSTRATION OF PLAYER AND ENEMY POSITIONS (VIEWED FROM ABOVE)

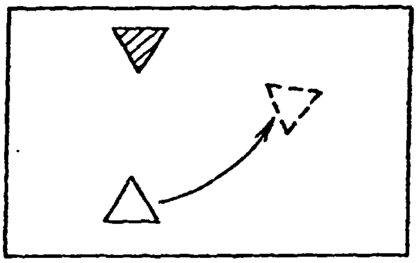
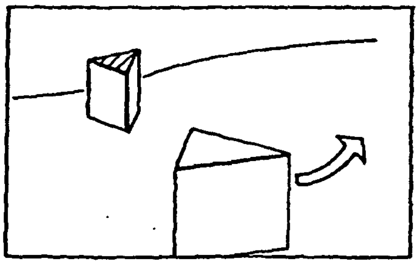
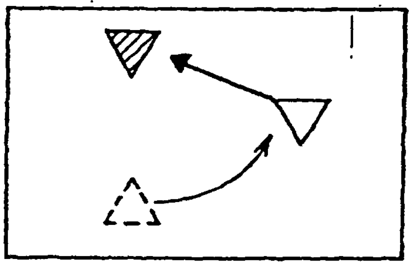


FIG.6C



VIRTUAL IMAGE DISPLAY FOR SCENE 1

FIG.6D



SCENE 1: ILLUSTRATION OF PLAYER AND ENEMY POSITIONS (VIEWED FROM ABOVE)

FIG.7A SCENE 2: ILLUSTRATION OF PLAYER AND ENEMY POSITIONS (VIEWED FROM ABOVE)

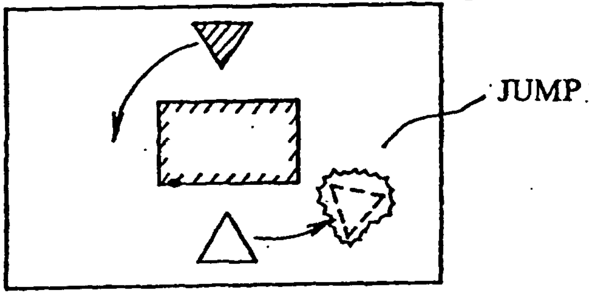


FIG.7B VIRTUAL IMAGE DISPLAY FOR SCENE 2

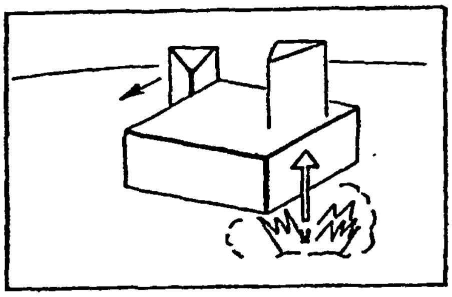


FIG.8A DRIVING GAME INPUT DEVICE

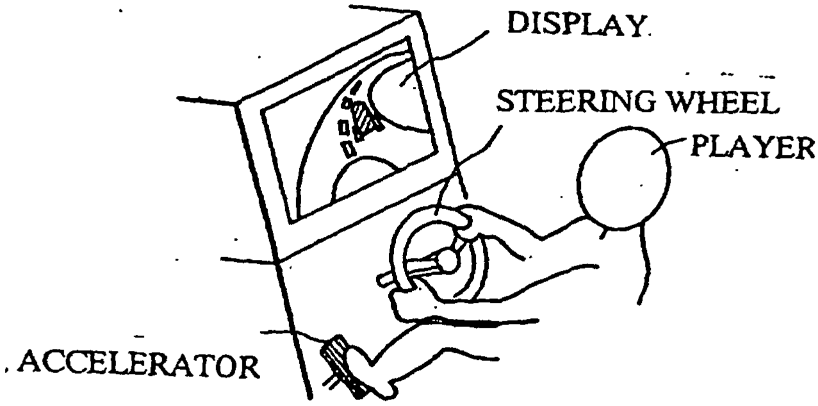
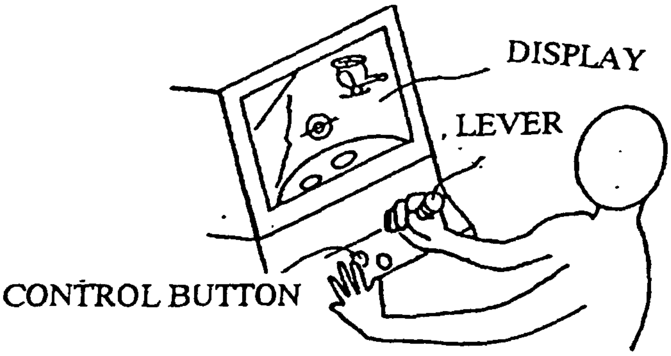


FIG.8B HELICOPTER SIMULATOR INPUT DEVICE


DISPLAY
LEVER
CONTROL BUTTON